THE UNIVERSITY *of* EDINBURGH

Signal Modelling Based Scalable Hybrid Wi-Fi Indoor Positioning System

Usman Syed

Doctor of Philosophy

THE UNIVERSITY OF EDINBURGH

2018

Abstract

Location based services (LBS) such as advertising, navigation and social media require a mobile device to be aware of its location anywhere. Global Positioning System (GPS) is accurate outdoors. However, in case of indoor environments, GPS fails to provide a location due to non-line of sight. Even in cases where GPS does manage to get a position fix indoors, it is largely inaccurate due to interference of indoor environment. Wi-Fi based indoor positioning offers best solution indoors, due to wide usage of Wi-Fi for internet access. Wi-Fi based indoor positioning systems are widely based on two techniques, first Lateration which uses distances estimated based on signal properties such as RSS (Received Signal Strength) and second, Fingerprint matching of data collected in offline phase. The accuracy of estimated position using Lateration techniques is lower compared to fingerprinting techniques. However, Fingerprinting techniques require storing a large amount of data and are also computationally intensive. Another drawback of systems based on fingerprinting techniques is that they are not scalable. As the system is scaled up, the database required to be maintained for fingerprinting techniques increases significantly. Lateration techniques also have challenges with coordinate system used in a scaled-up system. This thesis proposes a new scalable positioning system which combines the two techniques and reduces the amount of data to be stored, but also provides accuracy close to fingerprinting techniques. Data collected during the offline/calibration phase is processed by dividing the test area into blocks and then stored for use during online/positioning phase. During positioning phase, processed data is used to identify the block first and then lateration techniques are used to refine the estimated location. The current system reduces the data to be stored by a factor of 20. And the 50^{th} percentile accuracy with this novel system is 4.8m, while fingerprint system accuracy was 2.8m using same data. The significant reduction in database size and lower computational intensity benefits some of the applications like location-based search engines even with slightly lower performance in terms of accuracy.

Declaration of Originality

I hereby declare that this thesis has been composed by me and that except where stated, the work contained is my own. I also declare that the work contained in this thesis has not been submitted for any other degree or professional qualification except as specified.

24/05/2019

Usman Syed

Acknowledgements

Firstly, I would like to express my gratitude to my supervisor, Professor Tughrul Arslan for his continuous support throughout this study and research and for his guidance, motivation and advice.

I would also like to express my gratitude to the team at sensewhere Limited for providing access to tools and data used in this study. I would also like to thank Dr. Ahmet T. Erdogan and other members at System Level Integration Group (SLIG) for their support.

Finally, I would like to thank my parents, sister, wife and friends for their support and encouragement. I dedicate this thesis to their love, patience and belief in my capabilities.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

1.1 Indoor positioning

"Where am I?" is the question which has had a huge impact on the human civilization. During the early years of science, trying to answer the same question, Copernicus has changed the perception of Earth being the centre of the universe. Science and technology have moved on leaps and bounds since then. Answering the same question with new knowledge and from different contexts has resulted in some exciting discoveries and inventions. Knowledge about the stars helped early sailors to determine their position and navigate the oceans. New inventions have changed the technology used to answer the question "where am I?".

Radio navigation is the basis of the modern positioning systems. The initial usage of radio waves for navigation is LORAN (Long Range Navigation). LORAN was largely used for navigation on ships. This system was eventually replaced by the now well know Global Positioning System (GPS) which was originally developed for military purposes. In 1996 GPS was made available to the general public. GPS is a constellation of 32 satellites which can provide position anywhere in the world. Other similar satellite positioning systems have either already been deployed or in the process of being deployed. Some of these are GLONASS (Global Navigation Satellite System) developed by Russia, Galileo positioning system developed by European Union, Indian regional navigation satellite system developed by India and

Beidou navigation satellite system developed by China. When GPS (Global Positioning System) was made available to the general public, it was widely integrated into mobile devices such as mobile satellite navigation systems and Smartphones.

Smartphones have recently become ubiquitous and are embedded into the day to day life of most human beings. Smartphones provide an unprecedented access to the internet at our fingertips and is the most used feature on smartphones. According to the statistics [121], mobile phones now contribute to 51% of the internet traffic worldwide compared to less than 5% in 2009. The large increase in mobile traffic triggered market for context-based services. Some of the context factors that could be taken into consideration when providing services are individual's search history, Demographic profile, Time of the day, Weather, mood of the individual and Location. Location is one of the most critical factors in enabling some of the context aware services which are sometimes referred to as Location Based Services (LBS).

According to [123] 90% of the time is spent indoors by an individual at home, work and other indoor environments (such as restaurants and shopping malls). To support LBS, the mobile device had to be aware of its location anywhere and anytime. However, GPS cannot provide a location in all areas. In urban areas with high raised buildings, mobile device does not have a clear line of sight to satellites and the weak signal strength results in a bad or no GPS fix. In case of indoors, the GPS performance is particularly bad due to no line of sight at all. Some of the LBS use cases for indoor environment are discussed in following sections.

1.1.1 Location based search engine

According to a report published by Office for National Statistics [122], "*In 2017, the most popular internet activity was sending or receiving emails (82% of adults), up 3 percentage points from 79% in 2016. Finding information about goods and services was the second most popular at 71% of adults, up from 58% in 2007*". These findings indicate that smartphones are increasingly and predominantly being used for searching information on the internet. The search engines are transforming into personal assistant apps such as "Siri" on iPhones from Apple, "Google now" on Android phones from Google, and "Alexa" from Amazon. Multiple users making same query would have different reasons and goals and expect different results

depending on the context factors. Hence, the results are meaningful and helpful to the user when search engine is aware of the context. Location of the user is one of the most important factors which could provide the context when returning search results.

Since 90% of the time is spent indoors by an individual [123] and limitations of GPS indoors, indoor positioning systems play a key role in providing the Location parameter for context-based search. For example, Consider the following scenario where the user is indoors and searching for nearby pharmacies using the word "Pharmacy". Availability of location of the mobile device almost instantaneously (a few milli seconds) would enable the search engine to return a list of nearby pharmacies efficiently.

1.1.2 Location based advertising

Billions of dollars are spent on advertising worldwide and is also significantly growing [124]. With the increase in smartphones usage, more of this advertising is being targeted at the mobile devices, rather than traditional forms of advertising like billboards and television. Every time, an individual watches video online or uses social networking apps, they are targeted by advertisements. These advertisements can be tailored for each individual based on the context factors detailed in the previous section. In the current advertising use case, Location can play a vital role in customization of the adverts. A customised advertisement increases probability of interaction with the advertisement [125] by the user. For example, consider the case where the user is browsing on the phone and sees an advertisement for a discount on coffee at Starbucks. If user is near the store at the moment when the advertisement is displayed on the screen, the chances of user utilising the offer increases significantly. The device location should be estimated quickly and not drain the battery every time the position is updated before displaying a new advertisement.

1.1.3 Location based authentication

Availability of smartphones with access to internet has also paved way to mobile payments through mobile wallets such as "Apple pay" and "Google Pay". According to [127], "*The research shows that 17 percent of U.S. consumers now regularly use their smartphone to pay, up from 6 percent in 2014 when the survey was last conducted. In Europe, Spanish consumers are the most active users of*

mobile wallets, with 25 percent using them regularly, followed by Italy (24%), Sweden (23%) and the U.K. (14%).". A steady increase in usage of mobile wallets also has seen an increase in mobile payment frauds [126]. One of the solutions is to obtain device location at the point of payment and carry out verification based on the previous transaction history. For example, consider a user had completed a transaction at a store and then, within a couple of minutes, a transaction was made a few kilometres away. Using location information during the transactions can identify or tag the second transaction as spurious. This requires the mobile device to be able to obtain the location within a few milli seconds (almost real-time) so that, location estimation does not add significant overhead to the transaction time.

1.1.4 Indoor Navigation

Large indoor venues like Airports, shopping malls are getting bigger and need guidance to navigate these large indoor environments. Consider a user who is at a shopping mall and need to find a particular store. Similar scenarios are encountered at Airports, Railway stations and large public buildings. In these scenarios, an application running on a smartphone aware of its location can provide real-time navigation instructions. The location has to be continuously updated on the mobile device as the user moves in the indoor environment.

1.2 Motivation

The aim of the current work is to build an indoor positioning system which provides a solution for the scenarios detailed in the previous section. The following section summarises the current indoor positioning systems and identify the problems that the current work attempts to solve.

1.2.1 Summary of indoor positioning systems

All the scenarios described earlier require the device to be aware of its location in the indoor environments. There are various systems which have been developed based on different technologies to provide location in areas where GPS does not work. Some of the positioning systems are Active Badge [1] using the infrared signals, LANDMARC [2] using the RFID technology and CRICKET [3] using RFID and Ultrasound technology. These systems have a disadvantage of the need to install additional hardware in a building. An alternative was to develop systems which use existing infrastructure for positioning a mobile device indoors.

Some of the existing infrastructures that are used in positioning systems are mobile television signals [4], FM radio signals [5], phone networks [6] and wireless LAN signals [7]. A comparison of indoor positioning technologies and the accuracies is shown below in figure 1.1 [128].

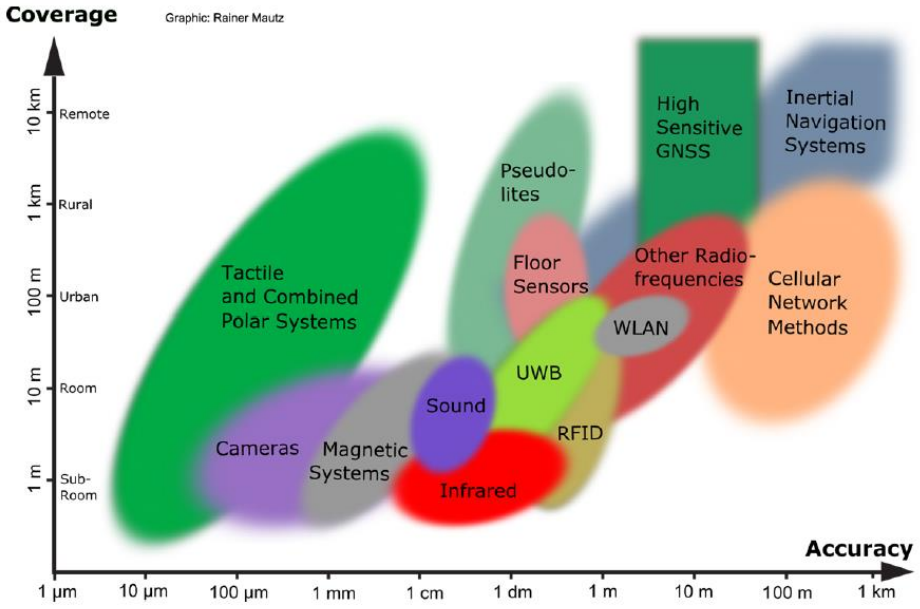


Figure 1.1 Comparison of indoor positioning technologies

The most popular indoor positioning systems, using existing infrastructure for mass market deployment are based on using the WLAN (IEEE 802.11 specification) operating in 2.4GHz frequency band. WLAN Access Points (APs) are widely used for wireless internet infrastructure and moreover, the mobile device does not have to actually establish a connection with the APs to identify them. The APs broadcast information such as its MAC address, the channel on which the AP transmits data, etc. The signal strength (SS) of the received signal at the mobile device can be used for positioning the mobile device [7]. Other Wi-Fi signal properties can also be used for positioning purposes such as SNR (Signal to Noise Ratio) [8]. There have been various techniques developed based on the RSS (Received Signal Strength) which can be broadly divided into two categories: first are the techniques based on distance estimation using RSS (Lateration systems) [9] and second, to match the RSS fingerprint (Fingerprint systems) [8] [10]. Fingerprint systems are also referred to as Scene analysis systems. Fingerprint systems provide best accuracy whereas Lateration systems are easier to implement. A comparison of some of the current positioning systems is presented in [11] [12] and [13].

1.2.2 Drawbacks of current Wi-Fi based indoor positioning systems

Both Fingerprint systems and RSS based distance estimation systems have their own pros and cons. This section summarises the drawbacks of each system.

1.2.2.1 Drawbacks of RSS based systems

RSS based techniques rely on estimating the distance between beacon and the mobile device reliably. The transformation of RSS to distance is usually based on propagation model which describes the variation of RSS with distance [52]. The model used significantly determines the accuracy of the overall system. But there is no single propagation model which can work for all buildings. Multiple propagation models have been proposed and used depending on the environment some of which are discussed in [55] to [58]. There are various factors which complicate modelling a generic propagation model. Some of these factors are floor layout, people moving in the building, time of the day and no line of sight due to walls cause severe signal attenuation and multipath. This makes it impossible to model a radio propagation which would work accurately in all buildings.

A distance estimation based indoor positioning system requires the location information of beacons. This requires either surveying test area to identify the installed beacons or obtain beacon location data during installation of infrastructure. Beacon's location data collection needs to be simplified and automated for a system to be scalable.

Challenges with distance estimation techniques:

1. No generic path loss model to accurately describe radio propagation in all indoor environments

3. Automation of beacon location data collection

1.2.2.2 Drawbacks of scene analysis-based systems

Another popular Wi-Fi positioning technique is location fingerprinting. Location fingerprinting techniques do not involve estimating the distances to the access points. Fingerprinting techniques [20] to [25] usually involve two phases. First is the calibration phase during which data is collected to create a radio map. The radio map contains the measured RSS values at multiple locations. Fingerprinting

techniques involve comparing the RSS values in the position request to the radio map generated during the calibration phase. The data collected needs to be stored and the memory required can be very large depending on the size and number of venues and the amount of calibration data at each venue. The algorithms to compare the position request with the RSS values in the memory are computationally intensive and time consuming.

A comparison of Wi-Fi based positioning systems along with other indoor positioning systems was presented in [11] and [135]. A summary of the Wi-Fi based positioning systems is summarized in the table 1.1 [11].

System/ Solution	Wireless technologies	Positioning algorithm	Accuracy	Precision	Complexity	Scalability/ Space dimension	Robust-ness	Cost
Microsoft RADAR	WLAN, Received Signal Strength (RSS)	*K* NN, Viterbi-like algorithm	3~5m	50%within around 2.5 m and 90% within around 5.9 m	Moderate	Good /2D,3D	Good	Low
Horus	WLAN RSS	Probabilistic method	2m	90% within 2.1m	Moderate	Good/2D	Good	Low
DIT	WLAN RSS	MLP, SVM, etc.	3m	90% within 5.12m for SVM; 90% within 5.40m for MLP	Moderate	Good/2D,3D	Good	Low
Ekahau	WALN RSSI	Probabilistic method (Tracking-assistant)	1m	50% within 2m	Moderate	Good/2D	Good	Low
Robot-based	WLAN (RSS)	Bayesian approach	1.5 m	Over 50% within 1.5m	Medium	Good/2D	Good	Medium
MultiLoc	WLAN (RSS)	SMP	2.7 m	50% within 2.7m	Low	Good/2D	Good	Medium
TIX	WLAN (RSS)	TIX	5.4 m	50% within 5.4m	Low	Good/2D	Good	Medium

Table 1.1 Comparison of Wi-Fi based positioning systems

For the systems summarized in the table, most of the systems use fingerprint-based positioning algorithm and report an accuracy of 1m-5m. A similar comparison is presented in [135], which includes systems integrating Wi-Fi algorithms with IMU (Inertial Measurement Units). Based on the works in [131], [132] and [135] accuracy of Wi-Fi algorithms varies between 2m-7m.

An analysis of the performance of multiple fingerprint based positing algorithms can be found in [130]. Computing time for three positioning algorithms was evaluated, which showed that neural network algorithm was the fastest and k-Nearest neighbour algorithm performed the worst with 100 samples. A table of the findings from [130] is shown below.

Positioning Algorithm	Computation time (sec)
Probabilistic Method	2
k-Nearest neighbour algorithm (With 100 samples at each location)	10
k-Nearest neighbour algorithm (With 10 samples at each location)	1
Neural networks	0.25

Table 1.2 Comparison of fingerprint algorithms and computation time

Depending on the application scenario, some of these algorithms with large latency in location estimation are not desirable. Some of the recent research, [133], [134] and [136] focuses on addressing the computational intensity and database size by using the Gaussian Process Regression which can reduce the system latency compared to Nearest neighbour algorithms. But are still computationally intensive compared to Lateration algorithms.

Challenges with finger printing techniques:

1. Need a large database to store the data collected during training/calibration phase
2. Computationally intensive when dealing with a large database

1.2.3 Research Objectives

The current research aims to develop an indoor positioning system that can support the scenarios described earlier, on a mobile device. When developing a positioning system, the following parameters should be considered: Accuracy, Costs, Coverage, System latency, Continuity and Number of users. Each of these parameters determine the scenarios a positioning system can support and the experience that can be delivered to the users.

A lot of the research being done in the field of indoor positioning is focused on the accuracy rather than how effective the system can perform in the real-life scenarios. [137] and [138] point out that current focus on the accuracy of a positioning system and also that the results obtained are from a controlled environment. Most of the scenarios described earlier can be solved with an accuracy

between 2m to 10m, except for the indoor navigation scenario where the position would have to be accurate up to 1m. The current work aims to solve the scenarios of Location based search, Advertising and Authentication.

The current work aims to keep the costs for deploying and maintaining the positioning system to minimum. This can be achieved by using the existing Wi-Fi infrastructure and use Bluetooth beacons in areas where existing Wi-Fi infrastructure coverage is poor. As both Wi-Fi and Bluetooth beacons are readily detected and scanned (along with RSS) by smartphones, there would be no additional costs incurred to make the hardware usable by smartphones. The costs to be considered are resources for data collection during offline phase, maintenance of database and resources to update the offline data.

Coverage is another important parameter to consider when developing a positioning system. Most of the systems use local coordinate system (local x and y co-ordinates) when estimating the location. When the system is scaled up to cover a city or a country or the whole world, using Geographical co-ordinates to display the location on a world map makes it easier. However, using geographical coordinates when dealing with tasks like calculating distance between two coordinates become computationally intensive. Works like [140] and [141] propose techniques to convert GPS coordinates to local range and bearing for simplifying dealing with GPS coordinates. This is usually done by converting GPS coordinates to ENU (Easting, Northing, Up) or ECEF (Earth Centred Earth Fixed) coordinates. The current work in this thesis aims to make this conversion to and from GPS co-ordinates simpler and computationally less intensive. As this conversation process is intended to be done on a mobile device, a less computationally intensive process would reduce the power consumption on mobile device.

System latency determines the types of use cases that can be supported by a positioning system. Consider the three scenarios, Location based search, Location based advertising and Location based Authentication. When a user is searching for information, the results are expected to be returned instantaneously. For example, Google even displays the time taken to return the results for every search. Adding location context to searches should be seamless and add minimum overhead to the overall time to return the results. Hence, system latency should be minimum even for

other scenarios like Location based authentication and advertising. [130] gives computation times for each fingerprint algorithm and [133], [134], [136] discuss methods to reduce computational complexity to reduce overall system latency.

Another factor to consider is continuity, where the system either provides a continuous position or a single position whenever requested. For use case of "indoor navigation", continuous position is required to update the route recommended to the user. [142] and [143] propose methods to integrate Wi-Fi based positioning with PDR (Pedestrian Dead Reckoning) to provide a continuous position. However, for the other scenarios being considered, returning a single position fulfils the requirement.

Depending on the architecture of positioning system, the number of users that can be supported varies. For example, consider a server-based system where the mobile device makes a request to the server and based on the position request the server makes an estimate, and returns the position to mobile device. In a server-based system, the server has to be powerful enough to process multiple simultaneous position requests from users. The algorithms used should be capable of keeping the system latency to the minimum for all users of the system. Complicated algorithms mean adding more resources for the server and there by higher costs to maintain the system.

Other parameters to consider is power consumption which is critical when using mobile devices. [144], [145] and [147] investigate into the trade-off between power consumption and accuracy. [144] rightly suggests that all applications do not need the best accuracy at all time and goes on to propose methods to use specific positioning algorithms based on the requirement. For better accuracy, other components on the mobile device like Compass, accelerometer are used for PDR, which contributes to higher power consumption. Even with server-based systems, the amount of data transferred between mobile device and server along with frequent usage of network component on the mobile device incurs higher power consumption [145]. One possible solution to avoid frequent network requests is to store data for a venue (or a nearby area) on the device, along with a simple positioning algorithm. This also helps the mobile device to estimate a position even in areas with no network coverage.

The aim of the current work is to develop an indoor positioning system which (is)

1. Requires minimum data storage and minimum computational intensity thereby has minimum latency
2. Provides accuracy between 2m to 10m to support the use cases discussed earlier
3. Scalable that can be deployed for large areas like cities covering multiple indoor areas
4. Capable of supporting large number of users
5. Capable of running on a mobile device
6. Does not require additional hardware installation

1.3 Limitations of Scope

The aim of current work is to develop an indoor positioning system that can work with mobile devices. Some assumptions are made to the limit scope of current work. The current system only considers static scenarios where the device makes a single position request and works with that single position request (without knowledge of historic locations). Since the current work targets single position request scenarios, hybrid technology systems, such as the ones which integrate Wi-Fi and PDR systems are not considered. It is assumed that, an existing Wi-Fi infrastructure is available, so that no additional hardware installation is needed. Also, the aim is not to develop the best positioning system in terms of accuracy but to develop a system which reduces power consumption, database size and system latency. The current work does not consider the impact of the make of Wi-Fi routers, as the system aims to use existing infrastructure and intends the system to work anywhere. The current system also, does not consider the effect of heterogenous devices on the system.

1.4 Contribution

The current work described in this thesis is an implementation of a novel hybrid indoor positioning system which combines fingerprinting and lateration techniques. The aim is to reduce the data stored and computational intensity of the system. At the same time make the system scalable and achieve accuracy (2m to 10m) that can support the use cases described earlier.

As part of the proposed hybrid system, the data collected during calibration phase is divided into blocks and a propagation model for each block is generated, before saving to a database. This approach has reduced the amount of data to be stored by about 20 times compared to a fingerprinting system. The impact of various block sizes on distance estimation is studied. And possibility of identifying the block size automatically given the calibration data has been evaluated. Lateration with spherical coordinates (x, y, z) use the height component (z) which is measured from centre of the earth. Three dimensional coordinates introduce further complexity when solving lateration equations. A transformation algorithm is proposed in this thesis to convert Geographic (latitude-longitude) coordinates to local two-dimensional Cartesian coordinates which makes trilateration less computationally intensive. Using two dimensional local coordinates reduces the number of iterations by 57% compared to three dimensional coordinates.

The location of the Wi-Fi access points is required for estimating the location of mobile device in the proposed hybrid system. This information is usually obtained from the department which installs Wi-Fi infrastructure in the building. This process can be automated by installing Bluetooth devices which broadcast the Wi-Fi access point's location information. The location information can be easily obtained by a mobile device by reading the data from the Bluetooth Beacons. Also, these beacons can be modified to be installed in areas with poor or no Wi-Fi infrastructure.

The work presented in this thesis has led to two publications and the references are attached in the appendix.

1.4.1 Summary

Proposed hybrid indoor positioning system was successfully implemented, which reduces data required to be stored by a factor of 20, compared to fingerprinting systems. The accuracy of estimated location from the implemented hybrid system was 4.8m at 50^{th} percentile and 11m at 90^{th} percentile compared to 2.8m at 50^{th} percentile and 6.3m at 90^{th} percentile from fingerprint systems.

Coordinate transformation algorithm which converts latitude-longitude to a two-dimensional local Cartesian co-ordinate system has reduced number of iterations for lateration process by 57%.

Bluetooth device was successfully implemented, which can broadcast Wi-Fi beacon location information and can be received by a mobile device with Bluetooth 4.0 capability.

1.5 Thesis structure

The current thesis is organised as follows:

Chapter 2 provides background about indoor positioning and literature review covering existing positioning systems and techniques. The major Wi-Fi based positioning techniques and their drawbacks are also discussed.

Chapter 3 describes the proposed Wi-Fi based hybrid positioning system. The chapter also details an experiment at a university building which implements the proposed system. The data collection process and processing of the collected data are explained. The results are discussed towards to end of the chapter. A list of improvements is proposed to the system at the end. (Work detailed in this chapter are published in Appendix [1]).

Chapter 4 analyses the impact of block size on the distance estimation and aims to identify an optimal block size. Distance estimation errors with various block sizes are calculated and alternative block size criteria are evaluated. Towards the end of the chapter, automatic block detection from calibration data with clustering algorithms is evaluated.

Chapter 5 explains a transformation algorithm to convert Geographic coordinates (Lat, Lon) into a local two-dimensional co-ordinate system. Then two dimensional lateration using non-linear least squares algorithm is described. A comparison of lateration using three dimensional coordinates and two-dimensional coordinates is presented. Also presents an experiment at one of the university buildings. This experiment implements the modifications proposed in chapter 3. Towards the end of the chapter results from hybrid positioning system are analysed. The reasons for block misidentification and some large errors in the position estimations are investigated.

Chapter 6 concludes the work presented in this thesis and discusses current limitations of the hybrid positioning system. This chapter also describes future work to further improve and integrate new techniques to improve distance estimation.

Chapter 7 provides an introduction to the Bluetooth 4.0 protocol which can be used to design low energy beacons. These beacons can transmit the location of Wi-Fi access points, when installed with the Wi-Fi access point. The signal characteristics of a Bluetooth 4.0 beacon are analysed (Work in this chapter is published in Appendix [2]). This chapter also details an android application which can be used to validate the location information. (Chapter 7 moved to appendix)

Chapter 2

Fundamentals of Indoor positioning

2.1 Introduction:

The previous chapter gives an overview of various technologies used for indoor positioning. Positioning systems implemented using technologies like Ultrasound, radio frequency identification (RFID) [3] and Infrared (IR) [1] require installation of additional infrastructure in the building. The systems which require custom hardware to be installed are usually not scalable due to the costs incurred for installation and maintenance. Most of the research in the recent years in the field of Indoor positioning extensively use radio signals that are already available in a building. Radio signals are integrated widely in the devices used in everyday life. Some examples of these radio signals are the TV broadcast signals [4], mobile cellular signals [6], FM radio signals [5], Wi-Fi signals and Bluetooth signals [7]. These radio signals can usually propagate indoors and with appropriate hardware to receive these signals, can be used for positioning.

This chapter summarises the positioning techniques that use radio signals and provide a background for the research detailed in this thesis. The aim is to develop a system which is scalable and has minimum running costs. Since Wi-Fi is widely used for internet services, these radio signals are ideal to be used for indoor positioning. This chapter first provides an overview of various positioning techniques that use Wi-Fi radio signals. These techniques can be broadly divided into four categories,

Proximity, Angulation, Lateration and scene analysis. Some of the techniques used in research presented here are discussed in detail on following chapters.

2.2 Positioning Techniques

A general overview of a positioning system would be helpful in understanding various positioning techniques. A positioning system would generally have a device whose location is to be estimated and some fixed devices (Beacons) in the area of interest. Depending on the positioning technique, Data regarding the beacons is collected prior to estimating location mobile device. The basic requirement of mobile device is to be able to detect nearby beacons. In the context of this thesis the device whose position is to be estimated is sometimes referred to as mobile device or target device and the fixed devices are referred to as beacons or WAPs (wireless access points). Figure 2.1 classifies the positioning systems based on the techniques used, detailed in [14].

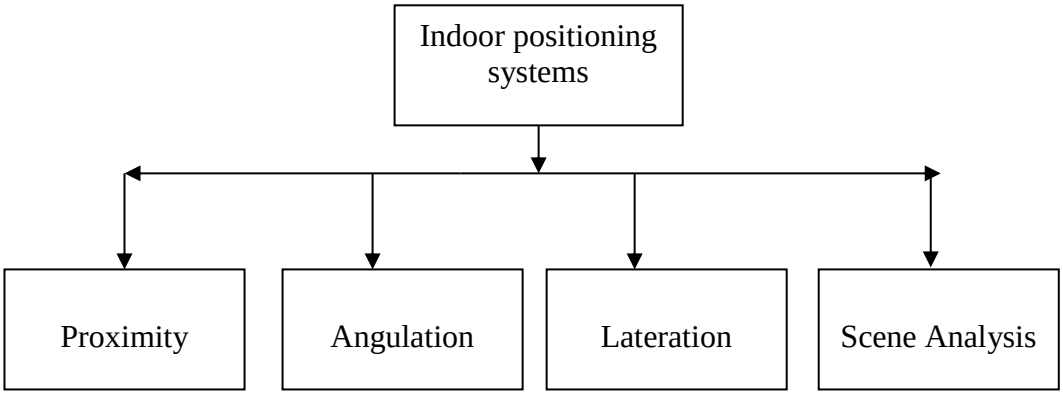


Figure 2.1 Indoor positioning system classifications

2.2.1 Proximity

Proximity technique estimates the location by detecting when the mobile device is within the range of a beacon. The position of the mobile device is estimated to be the location of the detected beacon. Figure 2.2 illustrates the basic principle behind proximity technique. The mobile device can either just detect the beacon or exchange data with the beacon to identify the location of the beacon. There are multiple implementations of this technique using classic Bluetooth and RFID technologies.

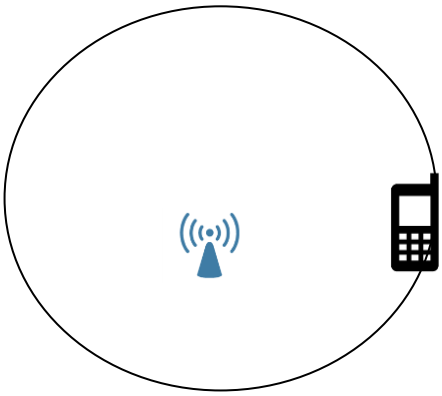


Figure 2.2 Proximity technique

In case of Bluetooth based system, Bluetooth beacons are installed in the area of interest. The location of the beacons can be stored as a database on a server or locally on mobile device [15]. Similar systems based on RFID, involve carrying a RFID tag whose location is identified when in proximity of a known RFID scanner [16].

RFID positioning systems like LANDMARC [2] and Spot On [149] use tags to identify the location by detecting the tags, when in range of a scanner. The tags operate at different frequencies varying from low frequency (124 – 135kHz) to high frequency (13.56MHz) to ultra-high frequency (860 - 960MHz). The range of these tags varies from less than a meter to tens of meters [148]. The accuracy of these RFID based systems is dependent on the read range. A comparison of various RFID positioning systems is presented in [150], which report accuracy of 1m to 3m.

The advantage of proximity technique is its simplicity. These systems do not require complicated algorithms and does not require maintenance of a large database.

However, there are disadvantages as well. These systems require installation of additional infrastructure and smartphones do not have RFID capability. Also, the accuracy is dependent on the range of the beacons. If the beacons have a long range such as the Wi-Fi access points which can have a range of up to 100m [151], the location of the mobile device can be anywhere within the 100m radius around the

beacon. Proximity systems using Wi-Fi infrastructure do not provide the accuracy to meet the requirements for the use cases set out in previous chapter.

2.2.2 Angulation

Angle of arrival (AoA) techniques estimate the location of a mobile device by using the estimated angle of arrival, of received radio signals. Algorithms are implemented based on number of beacons detected during location estimation. One of the scenarios is where two beacons are available and the angle to the mobile device from the beacons is known. In such scenarios, the location of the mobile device is determined by calculating the intersection of the lines originating from the beacons location as illustrated in figure 2.3.

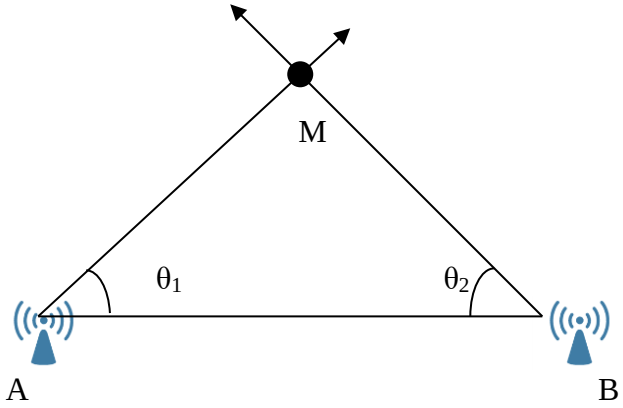


Figure 2.3 Angulation with two beacons

In figure 2.3, ‘A’ and ‘B’ are beacons whose location coordinates are known. The angles θ_1 and θ_2 are the estimated angles from beacons to mobile device. The location of mobile device ‘M’ is the intersection of lines originating from beacons ‘A’ and ‘B’ at angles θ_1 and θ_2 respectively. This technique can also be used in scenarios where more than two beacons are known. Figure 2.4 shows the scenario where location of three beacons (A, B and C) and corresponding angles (θ_1, θ_2 and θ_3) to the mobile device are estimated. Since estimation of angle of arrival can have errors, the lines do not always intersect. An approximate location is evaluated by calculating the intersection of lines originating from the beacons at corresponding angles. Some literature also uses the term Direction of arrival (DoA) for Angle of arrival (AoA). Some of the algorithms and techniques involved in AOA measurement are described in [17] and [18].

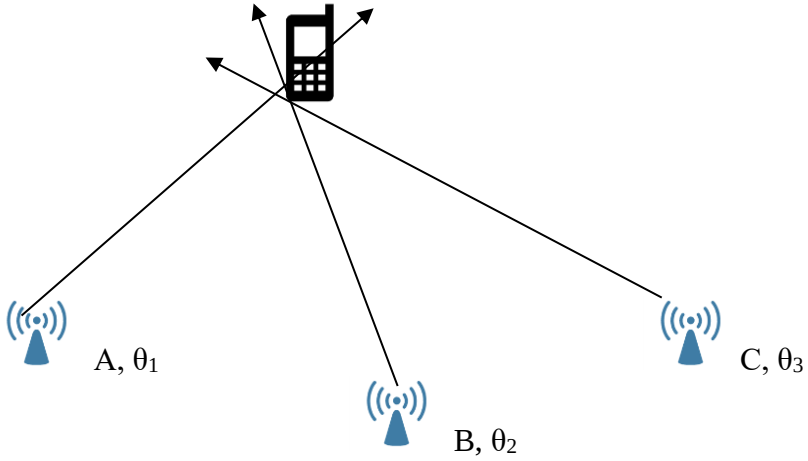


Figure 2.4 Angulation with three beacons

AOA, of an incoming signal can be determined by receiving it on an antenna array. By measuring the phase difference between the signals received on the antennas, the direction of the source is estimated [152]. However, in an indoor environment, signals from the same source can be received from multiple directions due to multipath effect. Due to multipath effect, the signal received with line of sight (LOS) to a reflected signal is hard to distinguish and difficult to estimate AOA.

The main advantage of Angulation techniques is that the mobile device and beacons do not need time synchronization. However, the angle of arrival measurement sometimes requires large and complex hardware which is hard to integrate into a mobile device. In an indoor environment, multipath reflections result in large errors in angle estimations. Work presented in [153] overcomes influence of multipath, by using a 16-array antenna. Research such as [154] and [155] aim to bring Multiple Input Multiple Output (MIMO) transceivers with Wi-Fi chips on mobile phones, which can make AOA estimation possible on mobile phones. But for now, AOA detection would require additional hardware which does not fit into the aim of current work.

2.2.3 Lateration

Lateration techniques estimate the location of mobile device by using the distance estimations to the beacons whose location coordinates are known.
The mobile device location is the intersection of circles with centre at the location of beacons and estimated distances as radii. In ideal scenario and a two-dimensional space, this technique requires at least three circles to identify a mobile device's location. This is illustrated in the *figure 2.5* which shows three beacons and the

circles with estimated distances to the mobile device. A, B and C are the beacons whose location coordinates are known and 'P' is the estimated location of the mobile device. R_a, R_b and R_c are the radii of the intersecting circles centred at the beacons (A, B and C). The intersection points P, P_1, P_2 and P_3 of the three circles are highlighted, of which P_1, P_2 and P_3 are the possible location of the mobile device where only two beacons are detected. Since three beacons were considered, this technique is also known as Trilateration. In a three-dimensional space, the location of mobile device is estimated by calculating intersection of three spheres. Position estimation in three-dimensional space require at least four beacons. The estimation of distance between the beacon and the mobile device is a critical part. The distance between beacon and mobile device is estimated using various techniques based on one of the signal properties (such as Timing, Phase and RSS). Some of these techniques are discussed in detail as part of section 2.4, as these techniques with Wi-Fi signals, meet requirements of the user cases discussed in chapter 1.

The main advantage of lateration techniques is that some of the Wi-Fi signal characteristics such as RSS are easily available on mobile devices. However, the accuracy of distance estimation depends on models used for transformation of signal properties to distance. Some of the lateration techniques are summarised in chapter 6 of [19].

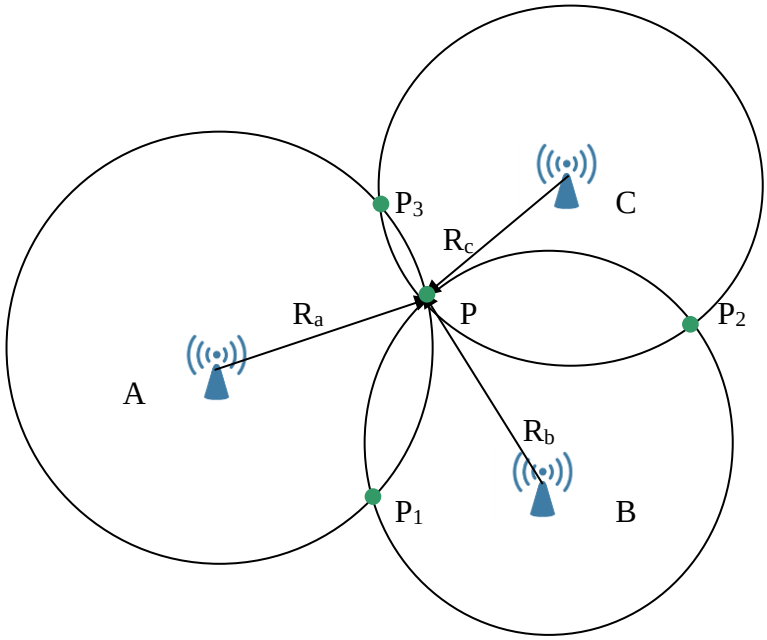


Figure 2.5 Lateration with three beacons

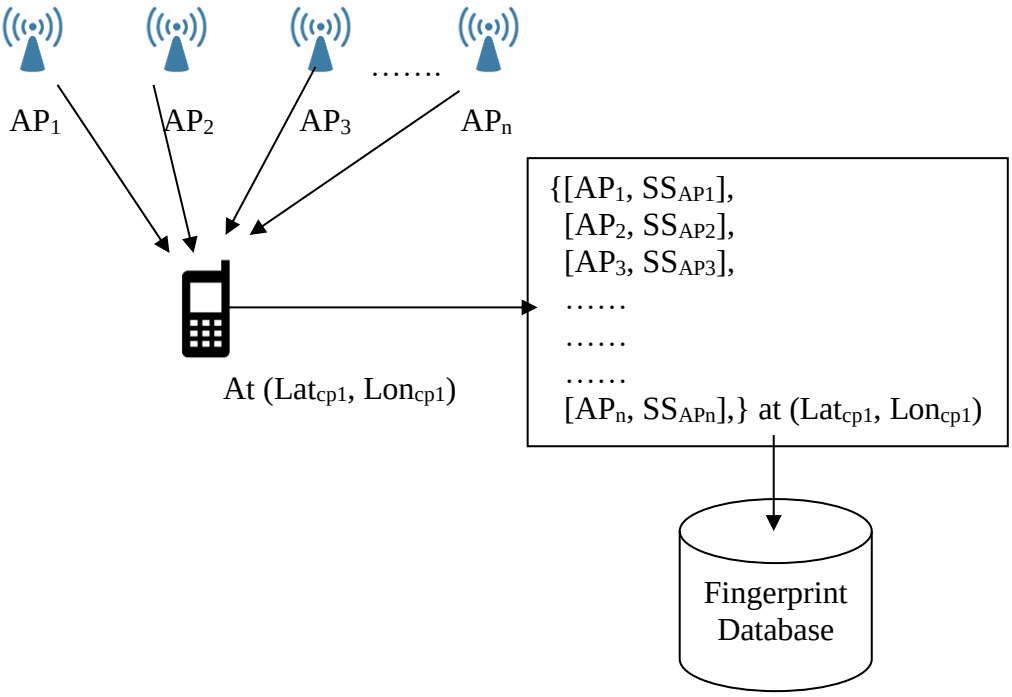
2.2.4 Scene Analysis

Scene analysis techniques typically have two phases. First, offline or training or calibration phase and second, online or positioning phase. During the calibration phase, radio signal features (such as RSS) also known as radio fingerprints are collected and stored. The data collected in calibration phase typically contains of location coordinates and signal strengths of beacons detected at the location. During positioning phase, the location of mobile device is calculated by matching request radio measurements from position request to radio fingerprints stored during the calibration phase. The following figure (Figure 2.6) provides a generic overview of Wi-Fi based scene analysis system.

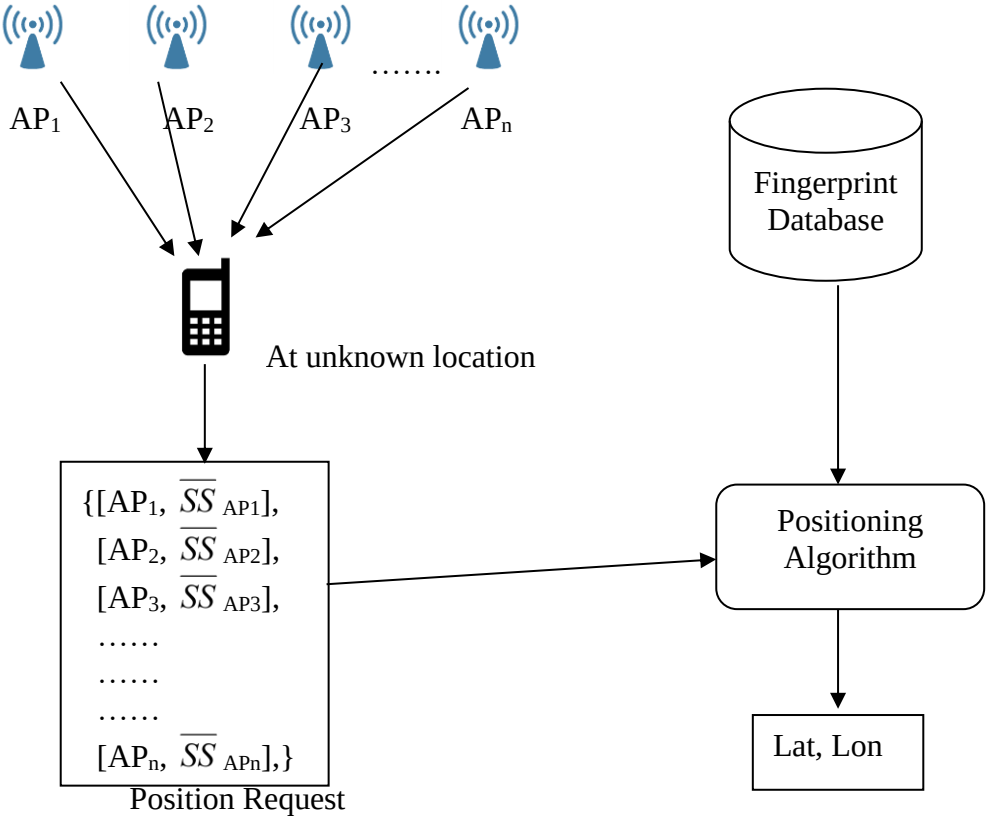
During calibration phase, mobile device at a known location (Lat_1, Lon_1) scans available Wi-Fi radio signals. The Wi-Fi beacons (AP_1, AP_2, AP_3… AP_n) along with corresponding RSS (SS_1, SS_2, SS_3… SS_n) are tagged with the known location and stored into a database. In some systems, the RSS values are measured over a period of time (for example 60sec) facing multiple directions and the average value is stored. This process is repeated at multiple locations; these locations are known as calibration points. Using the data collected at multiple calibration points, a fingerprint database (also known as radio map) is built as shown in figure 2.6(i).

During positioning phase, a mobile device scans for available Wi-Fi beacons and the corresponding signal strengths are obtained, which forms the position request shown in figure 2.6(ii). A position request is received as an input by the positioning algorithm. The positioning system compares the position request against the fingerprint database generated during calibration phase and estimates the location of mobile device.

There are various techniques which can be used to implement a scene analysis positioning system. Algorithms used in scene analysis systems can be broadly divided into two categories, Deterministic and Probabilistic methods. Some of these techniques are discussed in the next section. The main advantages of scene analysis systems are that they are accurate compared to lateration systems and can work with existing infrastructure. Also, when compared to the lateration systems, scene analysis systems do not need the actual location of radio signal sources. However, the data collection during calibration phase is time consuming and requires large resources.



i) Calibration Phase



ii) Online phase

Figure 2.6 Generic scene analysis system

2.3 Classification of scene analysis techniques

The general architecture of scene analysis positioning system was shown in figure 2.6. Scene analysis techniques differ from lateration based positioning systems significantly. These systems do not need actual location of the beacons for positioning and do not estimate distances to the beacons. Scene analysis systems use the fingerprint database generated during calibration phase to estimate the mobile device's location. There are various algorithms which can be used to compare the position request to the radio map. Algorithms used in scene analysis can be broadly divided into two categories, namely, Deterministic and Probabilistic methods.

In [7] a deterministic method system is proposed where, the average RSS for each Wi-Fi beacon is calculated at each calibration point to generate the fingerprint database. The RSS variation at a single point is large, which is further discussed in section 2.6. Probabilistic methods use RSS values for each Wi-Fi beacon to generate a probability distribution which considers the variations in RSS, instead of using an average of the observed RSS values [20].

2.3.1 Deterministic methods

In deterministic methods, the RSS readings from position request are compared with scans stored in the fingerprint database. The fingerprint database consists of calibration points which have corresponding average RSS values for each Wi-Fi beacon scanned, as illustrated in *figure 2.6*. The basic principle in all the deterministic methods is to calculate the signal distance between position request RSS vector $\{\overline{SS}_{AP1}, \overline{SS}_{AP2}, \overline{SS}_{AP3} \ldots \overline{SS}_{APn}\}$ from mobile device and RSS vectors $\{SS_{AP1}, SS_{AP2}, SS_{AP3} \ldots SS_{APn}\}$ from the database. The generalised signal distance can be described mathematically as follows

$$D = \left[\sum_{i=1}^{n} | \overline{SS}_{APi} - SS_{APi} |^{p} \right]^{1/p} \quad \ldots\ldots (2.1)$$

Where D is the signal distance between position request vector and RSS vector from the database

'n' is number of Wi-Fi beacons in the vector

The value of 'p' determines the distance calculated. If 'p' equals to '1' the distance calculated is known as Manhattan distance and if 'p' equals to '2', Euclidean distance.

Substituting ‘p’ =1 in equation (2.1), Manhattan distance is given by

$$D_{Man} = \sum_{i=1}^{n} | \overline{SS}_{APi} - SS_{APi} | \quad \text{…….. (2.2)}$$

Substituting ‘p’ =2 in equation (2.1), Euclidean distance is given by

$$D_{Euc} = \left[\sum_{i=1}^{n} | \overline{SS}_{APi} - SS_{APi} |^{2} \right]^{1/2} \quad \text{…….. (2.3)}$$

Multiple algorithms can be implemented based on the calculated distance. In nearest neighbour (NN) algorithm, the point which gives the shortest distance is returned as the location of mobile device. In ‘K’ nearest neighbour (KNN) algorithm, ‘K’ ($K \geq 2$) shortest distances are used for calculation of the location of mobile device. The average of location coordinates of ‘K’ points is returned as the location of mobile device [21]. In ‘K’ weighted nearest algorithm (KWNN), the weighted average is used to calculate the location of mobile device [22]. [156] provides accuracy results using NN, KNN and KWNN algorithms. Instead of averaging the signal strengths from scans at same location, from different orientations, [156] stores them as multiple fingerprints for same location. [156] considers multiple variations to the number of fingerprint locations considered for KNN and KWNN algorithms. The accuracy varies between 2m to 9m for all the variations considered.

[158] discusses the computational intensity when basic NN algorithm is implemented on a mobile device. The computational intensity of NN algorithm, is of the order, O(B*CP) where ‘B’ is the number of beacons and “CP” is the number of calibration points. It can be noticed that as the number of beacons and calibration points in the positioning system increases, the computational intensity increases. [157] implements the KNN algorithm on an android phone and concludes as follows “fingerprinting technique require approximately 2-3 seconds for 30-40 samplings. This latency is too high and is not suitable for real-world usage”.

However, Deterministic algorithms can still be applied to scenarios where, the number of calibration points is not as large to push the latency into seconds. By lowering the number of required calibration points, it is possible to estimate the coarse location of a mobile device. Once a possible region is identified, the location of mobile device can then be refined using other light weight algorithm such as

lateration. The current work considers deterministic fingerprint algorithms as one of the options to solve the user scenarios described in chapter 1.

2.3.2 Probabilistic methods

Bayes rule forms the basis of probabilistic methods. The data collected during calibration phase is used to calculate conditional probability of RSS vector (o_i) given the location (l_i) for all 'n' calibration points and is represented as $p(o_i|l_i)$. During positioning phase, the location (l) of mobile device is estimated given the RSS vector (o), represented as $p(l|o)$. Applying Bayes rule, the probability of location of mobile device is estimated as follows:

$$p(l \mid o) = \frac{p(o \mid l) * p(l)}{p(o)} = \frac{p(o \mid l) * p(l)}{\sum_{i=1}^{n} p(o \mid l_i) * p(l_i)} \quad \ldots\ldots (2.4)$$

Where $p(l)$ is probability of being at location 'l' prior to observing RSS vector 'o'

$p(o)$ is probability of observation 'o' which can be calculated as shown in (2.4) and can be treated as a normalising factor.

Some implementations of probabilistic methods can be seen in [24] and [25]. The probability of RSS vector (o) given the location (l) represented as $p(o|l)$ can be estimated using various methods such as Histogram method and Kernel method. A comparison of various probabilistic methods can be found in [26].

[159] describes alternative to the deterministic methods, where a probability distribution for each Wi-Fi beacon is calculated based on a set of samples at each location. The probability distribution is generated using histogram. Histogram, estimates the probability of a Wi-Fi beacon (AP_i) to have signal strength (SS), as frequency of occurrences of 'SS' over a total number of samples 'L', $P_{AP_i}(SS) = \frac{N_{APi}(SS)}{L}$ where $N_{APi}(SS)$ is the number of samples for signal strength is 'SS'. A mobile device only measures the RSS in integers which enables generation of histogram for each value of RSS. This process requires multiple readings for each Wi-Fi beacon at each location to be able to cover all the possible values of RSS at that location. Moreover, this process is repeated for all Wi-Fi beacons in the test area which increases the effort during offline phase. During online phase, multiple samples from the unknown location are used to generate probability distributions similar to offline phase. These probability distributions are then compared to the ones

generated during offline phase. This is done in [162] by calculating the Bhattacharya ratio for the strongest Wi-Fi beacons. [160] and [161] also propose similar histogram based probabilistic systems which aim to reduce the computational intensity in deterministic systems.

[25] presents a location estimation system using kernel based probabilistic method. In Kernel based methods, a probability mass is assigned for each of the samples in training data, described as a kernel. The probability of observation 'o' at location 'l' denoted as $p(o|l)$ is given by $p(o|l) = \frac{1}{n}\sum_{l_i=l} K(o, o_i)$ where $K(o, o_i)$is the kernel function, which usually is a Gaussian kernel represented as $K_{Gauss}(o, o_i) = \frac{1}{\sqrt{2\pi}\sigma} exp\left(-\frac{(o-o_i)^2}{2\sigma^2}\right)$ where σ is the parameter that determines the width of the kernel. In Wi-Fi positioning context, the formulae mentioned earlier can be replaced with an array, representing multiple beacons observed in each sample. In kernel-based methods, the calibration data is processed in two ways. First, all the samples from a single location are clustered and then probability function is generated. Alternatively, each sample is considered as independent observation and multiple probability functions are generated for each location. [25] favours treating each sample as independent observation and claims the second approach has produced better results.

Probabilistic methods estimate the location of mobile device by calculating probability of the RSS observation 'o_{unkn}' belonging to one of the locations from database. Observation 'o_{unkn}' consists of 'm' beacons, which can be represented for the observation as $o_{unkn} = \{f_1(\mu, \sigma) \dots f_m(\mu, \sigma)\}$ where $f_1(\mu, \sigma)$ represents the normal distribution of beacon '1' with mean 'μ' and standard deviation 'σ'. The location of the mobile device is estimated by calculating the probability of the observation belonging to each of the calibration points stored in the database. The probability is calculated by multiplying the probabilities of the RSS of each beacon belonging to a calibration point which can be represented as follows $\prod i = \sum_{j=1}^{m} P(o_j | f_j(\mu, \sigma))$ [156]. This probability is calculated at all locations and the location 'i' is returned as location of the mobile device if it returns the maximum value.

These probabilistic methods are also computationally intensive during the online phase, as the position request is matched with all the calibration points for the closest match. [161] presents a probabilistic system called HORUS, with an aim to lower the computational requirements and high accuracy. This system reduces the computational requirements by using a clustering module, which defines cluster as a set of locations which have common Wi-Fi beacons. [162] proposes a zone-based RSS reporting where a location server translates the RSS to a geographical zone which is used for Place of interest (POI) detection. Other ways to reduce the computation requirements and conserve power were proposed in [163], which limits the Wi-Fi scanning to only few channels. [164] and [165] propose computing and storing some of the variables in the offline mode, reducing the computational effort during positioning phase.

2.4 Distance estimation techniques for lateration

Lateration techniques described in the previous sections rely on estimating the distance to beacons. There are multiple techniques used for distance estimation in case of radio signals and figure 2.7 summarises various distance estimation techniques discussed in this section.

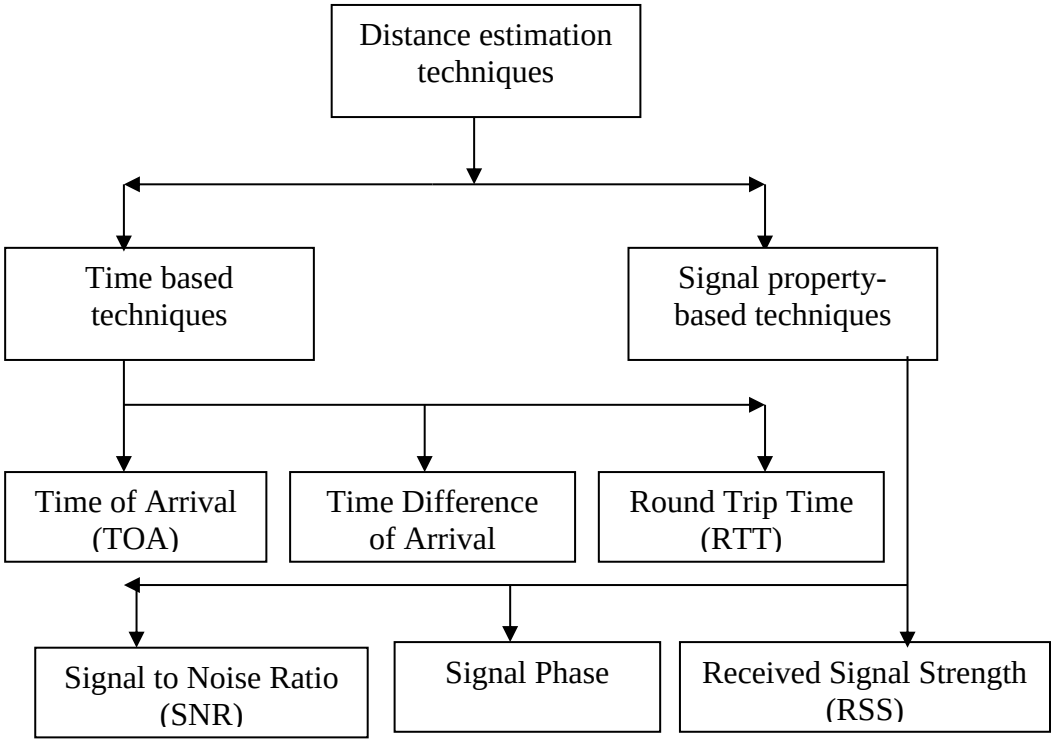


Figure 2.7 Distance estimation techniques classification

The distance between two radio devices such as the beacon and mobile unit is inferred from the signal parameters which vary with distance. The two parameters which vary with distance are time it takes to travel and signal properties (such as RSS) of transmitted signal. Distance estimation techniques can be divided into two categories, time based and signal property-based techniques. Time based techniques estimate time taken by the radio signal to travel from a beacon to mobile device. The distance is then calculated by multiplying the estimated time with speed of radio signal. Some systems use variation of radio signal properties over distance to estimate the distance between mobile device and beacons. Signal property variation is sometimes complex and needs to be modelled. These models of signal property variation are then used to estimate the distance between mobile device and beacon.

2.4.1 Time based distance estimation techniques

2.4.1.1 Time of Arrival (ToA):

Time of arrival (ToA) is also known as Time of flight (ToF). This technique uses the time taken for the radio signal to travel from beacon to the mobile device to estimate the distance. Radio signal data includes a time stamp when transmitted from the beacon and difference between the time received and transmitted gives time of flight. The speed of transmitted signal along with time of flight is used to calculate the distance between beacon and mobile device. The distance travelled by transmitted signal is given by product of time of flight and speed. Since the time is critical element in calculation of the distance, the beacon and mobile device need to be synchronised with a precise clock. The main disadvantage of this technique is the need to maintain a precise synchronisation between the mobile device and all the beacons. Moreover, In Wi-Fi signalling protocol, this precise time stamping of transmitted signals is not available. Some of the implementations of time of arrival techniques in other radio signal technologies are discussed in [27] and [28] in context of UWB (Ultra-Wide Band systems) which supports TOA. [166] proposes usage of ‘ACK’ (Acknowledgement) sequence in Wi-Fi (802.11) protocol data packets. However, this packet information is not readily available on a mobile device due to operating system restrictions. Due to these restrictions, TOA is not considered in the current work.

2.4.1.2 Time difference of arrival (TDoA):

Time difference of arrival (TDoA) technique measure the time delays between multiple beacons and the relative measurements are used for distance estimation. The advantage of this system is that, there is no need to maintain a precise synchronisation between the beacons and mobile device. But this still need the time stamping of signals transmitted which is not supported in Wi-Fi. A TDoA measurement method for Wi-Fi protocol was proposed in [29]. Measurement of TDoA requires additional hardware on the mobile device which is not intended as part of current work.

2.4.1.3 Round trip time (RTT):

Round trip time (RTT) also known as Round trip time of flight (RTToF), techniques measure the time taken for the signal to travel from the mobile device to the beacon and back. In techniques mentioned earlier, to calculate the time of flight, precise local clocks on both beacon and mobile device were required. However, in this technique a single clock on mobile device would suffice. The drawback of this technique is the time measurements have to be made simultaneously to multiple beacons. Since the mobile device is moving there would be latencies in connecting to multiple beacons, resulting in erroneous estimation of distances. Another drawback when using this technique with Wi-Fi beacons is mobile device needs to connect to the beacon which is not always possible. However, an algorithm to measure the RTT is proposed in [30], which cannot be implement on current mobile devices as the protocol is not currently supported. “802.11mc” [167], Wi-Fi protocol allows devices to measure the distance to nearby Wi-Fi beacons. However, Wi-Fi beacons which support this protocol are not available widely in the market. But when these Wi-Fi beacons are widely deployed, RTT measurement would be readily available and accurate distance measurements could be made.

All the time-based techniques described so far would need additional timing information from the radio signals or need to connect to the beacons. Hence these techniques are not feasible to use with the current Wi-Fi protocol. Since radio waves travel at speed of light (3×10^8 m/sec), any timing measurements would have to be precise to a Nano second. Even a Nano second error would mean approximately

0.3m error. Since the current mobile devices cannot support the protocol and precision in timing needed for distance estimation, these systems are not considered in the current work.

2.4.2 Signal property-based distance estimation techniques

2.4.2.1 Received signal strength:

The most widely used signal property for positioning is signal strength. The distance between the mobile device and a beacon can be estimated based on signal strength variation. A basic illustration of the signal strength variation to time is illustrated in figure 2.8. RSS at point A is stronger than at point B which is further away from the beacon.

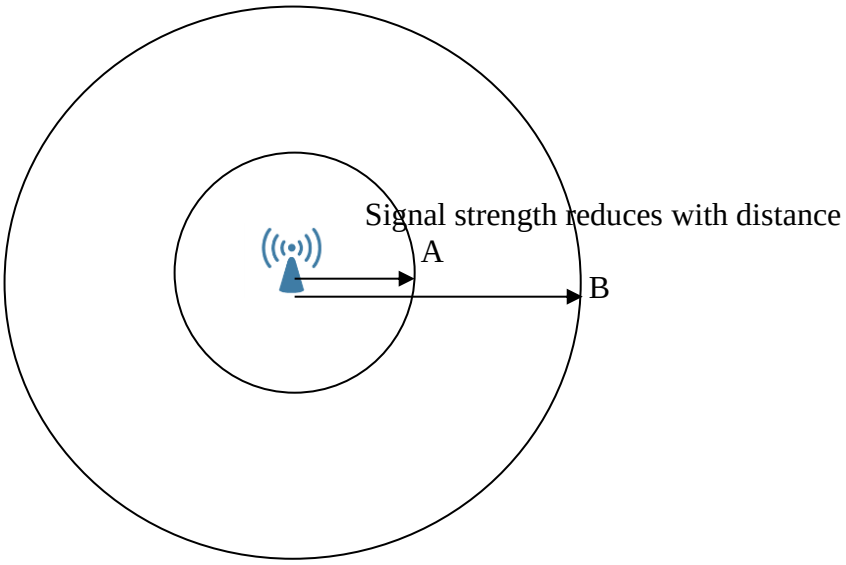


Figure 2.8 Received Signal Strength (RSS) variation with distance

Figure 2.8 can be considered as a generic RSS variation where the only loss is due to free space path loss. But, the radio propagation is complicated and difficult to predict indoors, where RSS does not vary uniformly around the beacon as illustrated in figure 2.8. Some of the environmental effects such as reflection, diffraction, scattering and multipath have a major impact on the wave propagation. A more detailed discussion of radio signal propagation is provided in section 2.6. One of the systems based on RSS is described in [34].

2.4.2.2 Signal phase

Signal phase methods use phase or sometimes phase difference to estimate the distance between mobile device and beacon. This method is also called as phase of arrival. Mobile device would require specific hardware to measure phase of received signal. However, phase measuring hardware is not readily available on mobile devices. Moreover, a mobile device needs Line of sight (LoS) to beacon to make an accurate phase measurement. In indoor environments, LoS is extremely limited leading to large errors in distance estimation. Some signal phase measurement techniques in wireless systems are discussed in [31]. [168] presents a VR (Virtual Reality) headset tracking system based on phase detection. The headset has multiple antennae to receive the same signal from a Wi-Fi beacons and there by identify the phase difference using MUSIC (MUltiple SIgnal Classification) algorithms to track the VR headset. Multiple antennae for Wi-Fi receivers are not available on mobile devices, hence these methods are not considered in current work.

2.4.2.3 Signal to noise ratio (SNR)

Signal to noise ratio is a measure of desired signal to background noise. This signal property has been used in some literature for positioning mobile device. SNR has been found to be less stable than RSS as described in [7] and [32]. Moreover, some Wi-Fi chipsets do not report SNR value as described in section 2 of [33], which makes it unsuitable for a scalable system.

2.5 Mathematical lateration solutions

The previous sections describe techniques to estimate the distances between a beacon and mobile device. The lateration process uses these estimated distances between multiple beacons and mobile device at an unknown location. A lateration problem is to find the intersection of circles in a two-dimensional space or in case of three-dimensional space, to find the intersection of spheres. For current discussion of lateration solutions, a three-dimensional space is considered.

Consider four beacons with location coordinates, (x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3) and (x_4, y_4, z_4) and distances to mobile device's location are r_1, r_2, r_3 and r_4. The lateration problem is to find intersection of four spheres with centres and radii described earlier. A sphere can be mathematically represented by equation (2.5).

$$(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2 = r_i^2 \ldots\ldots (2.5)$$

Equation (2.5) represents a sphere with centre at (x_i, y_i, z_i) and radius r_i. The mathematical background required for solving these equations are discussed in [35] and [36]. An implementation of three dimensional lateration is presented in [37]. There are three ways to solve a system of non-linear equations.

a) Linearization of the system of equations
b) Linear least squares
c) Non-linear least squares

2.5.1 Linearization of system of equations

The linearization reduces second degree equation (2.5) to linear equations. The linear equations in current three-dimensional space represent planes and if a two-dimensional space was being considered those linear equations would be representing lines. The linearization process involves adding and subtracting x_1, y_1 and z_1 from each sphere equation. This can be illustrated as follows.

$$(x - x_1 + x_1 - x_i)^2 + (y - y_1 + y_1 - y_i)^2 + (z - z_1 + z_1 - z_i)^2 = r_i^2 \ldots\ldots (2.6)$$

Expanding and re-arranging above equation

$$(x - x_1)(x_i - x_1) + (y - y_1)(y_i - y_1) + (z - z_1)(z_i - z_1) = b_{i1} \ \ldots\ldots (2.7)$$

Where $b_{i1} = \frac{1}{2}[r_1^2 - r_i^2 + d_{i1}^2]$

$$d_{i1} = \sqrt{(x_i - x_1)^2 + (y_i - y_1)^2 + (z_i - z_1)^2}$$

This linearization process is repeated for other sphere equations giving a system of linear equations shown below in matrix format.

$$Ax=b \ \ldots\ldots \qquad (2.8)$$

Where $A = \begin{bmatrix} x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ x_3 - x_1 & y_3 - y_1 & z_3 - z_1 \\ .... & ... & ... \\ ... & ... & ... \\ x_n - x_1 & y_n - y_1 & z_n - z_1 \end{bmatrix}$ $x = \begin{bmatrix} x - x_1 \\ y - y_1 \\ z - z_1 \end{bmatrix}$ and $b = \begin{bmatrix} b_{21} \\ b_{31} \\ ... \\ ... \\ b_{n1} \end{bmatrix}$

The system of linear equations is represented in the matrix form by equation (2.8). The following constraints apply to the equation 2.8 above. First, the location of beacons and estimated device's location are within the test area. Second, the estimated distances to the beacons ('r_i') are always positive. These equations have

three variables and hence would require at least four Wi-Fi beacons to calculate location of mobile device.

Solving equation (2.8) gives x which is $\begin{bmatrix} x - x_1 \\ y - y_1 \\ z - z_1 \end{bmatrix}$ and the location of the mobile device can be calculated by adding $\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}$.

However, there is a limitation to linearization due to the errors in distance estimations used as radii of spheres. The errors in distance estimation result in non-intersecting spheres and no solution for the system of linear equations can sometimes be determined.

2.5.2 Linear Least squares

The limitation in solving the linear system of equations due to errors in distance estimation can be solved using linear least squares. Linear least squares method is also helpful in the scenarios where the system of equations is over determined (number of equations available is more than required).

Linear least squares approach involves finding a solution which minimises error when the solution is used in system of equations described in (2.9). This can be illustrated mathematically as follows

$$r = \|Ax - b\|^2 = \sum_{i=1}^{m} (\sum_{j=1}^{n} a_{ij} x_j - b_i)^2 \quad \ldots\ldots \qquad (2.9)$$

Where ‘r’ is the residual or error

‘A’ is m×n matrix as illustrated in equation (2.8)

A value of ‘x’ for which residual ‘r’ is minimum is called least square solution. There are various algorithms for computing the linear least squares solution. Some of the linear least square algorithms are QR decomposition discussed in [38] and Single Value Decomposition (SVD) discussed in [39]. Some implementations and analysis of linear least squares applied to positioning are discussed in [40] and [41].

2.5.3 Non-Linear Least squares

Non-linear least squares algorithm is different to linear least squares in that the residual function ‘r’ is non-linear. When using non-linear least squares, the

equations of spheres do not have to be linearized. Some of the Non-linear least squares algorithms are discussed in [42] and some of the non-linear least square algorithm implementations in indoor positioning are discussed in [43] and [44]. The residual function (F_r) for the current scenario of four beacons in a 3-dimensional space can be described as follows.

$$F_r(x, y, z) = \sum_{i=1}^{n} (\sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2} - r_i)^2 \quad \ldots\ldots(2.10)$$

Non-linear least squares algorithm is an iterative process and needs an initial guess as input to the calculation algorithm. There are multiple algorithms to implement Non-linear least squares such as Gauss-Newton algorithm and Levenberg -Marquardt algorithm. The implementation of these algorithms is described in [45].

2.6 Properties of Wi-Fi Received Signal Strength

Received signal strength gives an indication of the distance from the beacon and can be used for distance estimation. But the measured RSS is affected by various factors some of which are related to the environment in which the beacon is being placed and hardware used to make the RSS measurement. A study of the Wi-Fi received signal strength is presented in [46] to [48]. To highlight the influence of the hardware, a simple experiment is described in the following section.

2.6.1 Variation of Wi-Fi Received Signal Strength over time

A Wi-Fi access point (Linksys WRT54 router) was placed in a classroom and RSSI values were collected using free software known as “inssider” running on a laptop. The software scans Wi-Fi access points and measures corresponding RSS which is logged into a file. The distance between the Wi-Fi access point and the laptop was 5m and RSS was measured every second. RSS for a single Wi-Fi access point (MAC id 08:76:FF:B3:2E:7B) was collected over one minute at a fixed point 5m from the access point. The log file was processed and RSS values were plotted as shown in figure 2.9.

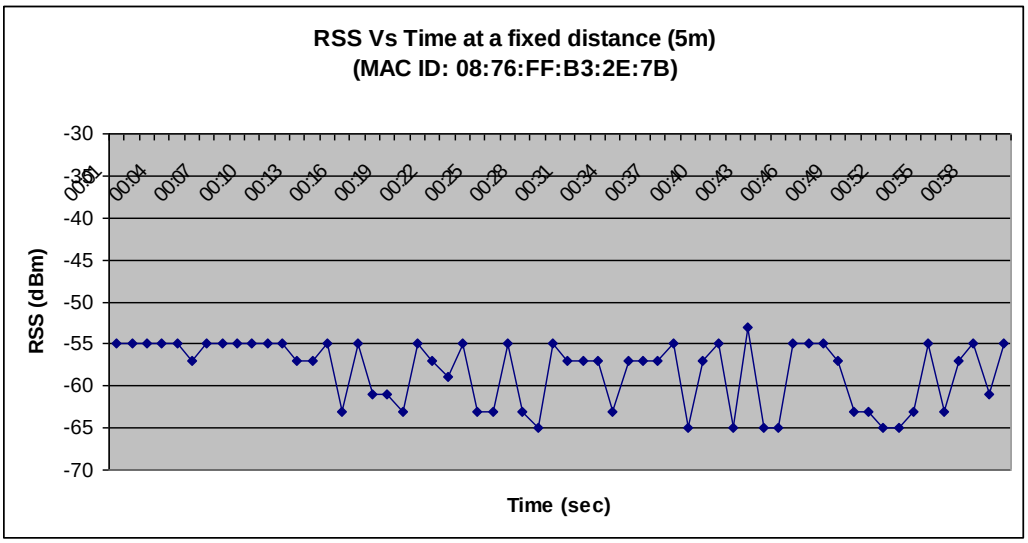


Figure 2.9 Variation of RSS over one minute at distance of 5m

It can be clearly noticed in figure 2.9 that RSS is not constant over the one-minute duration even though laptop was at a fixed point. This experiment shows that hardware used also affects RSS. RSS reading fluctuates up to 5dBm, even though there are no variations in the distance and environment. These fluctuations are due to the limitations in the hardware used in radio wave generation and reception.

User's body can also have an effect on the RSS if the user is between beacon and mobile device. Since the human body is 70% water, it absorbs radio signals from Wi-Fi beacons. Due to the user's body interference, the RSS is affected and there by the distance estimation used for position estimation. The effect of user's body, orientation is studied in [46]. Work presented in [46] shows that standard deviation of RSS due to user's body is approximately 0.68dBm to 3dBm and also a significant impact on RSS due to user's orientation. RSS measurements are also affected by the hardware used for measurement and this effect is studied in [47].

2.6.2 Variation of Wi-Fi Received Signal Strength with distance

The signal strength of a radio wave decreases as the user moves away from the source as illustrated in figure 2.8. A detailed background of radio propagation can be found in [49] to [51]. Friis transmission equation gives the amount of power received under ideal conditions such as no obstructions between the transmitter and receiver. Simple form of Friis transmission equation [51] is as follows,

$$P_r = P_t G_t G_r \left(\frac{\lambda}{4\pi R} \right)^2 \ldots\ldots (2.11)$$

Where P_t is power at transmitter

P_r is power at the receiver

G_t is transmitting antenna gain

G_r is receiving antenna gain

λ is wavelength of the radio wave

R is the distance between transmitter and receiver

If the antenna gain is assumed to be '1', equation (2.11) can be simplified as

$$P_r = P_t \left(\frac{\lambda}{4\pi R} \right)^2 \quad \ldots.. (2.12)$$

The equation (2.11) indicates that the power at receiver is inversely proportional to square of the distance.

$$P_r \propto \frac{1}{R^2}$$

This equation also forms the basis for free space path loss model. Equation (2.12) can be written as

$$\frac{P_t}{P_r} = \left(\frac{4\pi R}{\lambda} \right)^2 \ldots\ldots. (2.13)$$

In decibels, the above equation can be re written as

$$P_t(dB) - P_r(dB) = L_p = 10\log\left[\left(\frac{4\pi R}{\lambda} \right)^2 \right] \ldots.. (2.14)$$

Substituting $\lambda = \frac{c}{f}$ in equation (2.14) as in [49],

Where c is the speed of light approximately 3×10^8 m/sec

f is frequency of the radio wave measured in Hz (for Wi-Fi beacons, the operating frequency is 2.4GHz)

Equation (2.14) is simplified as

$$P_t(dB) - P_r(dB) = L_p = 40.04 + 20\log(R) \ldots. (2.15)$$

Equation (2.15) can be rewritten as

$$L_p = L_0 + 10\gamma \log(R) \ldots\ldots (2.16)$$

Where $L_0 = 40.04$

γ is referred as path loss co-efficient

R is the distance between transmitter and receiver

Equation (2.16) is the basic path loss model for free space environment. This model is used as basis for the modelling of various path loss models. Path loss models are further discussed in the next section.

2.7 Radio propagation models

A radio propagation model is mathematical representation of propagation characteristics as a function of frequency of the radio wave and distance between transmitter and receiver [52]. Since path loss is the most dominant characteristic of radio wave propagation, most models focus on generating a mathematical representation of the path loss. Path loss is the difference between transmitted power and received power in dB. The propagation model used for distance estimation based on RSS has a significant impact on the accuracy. Using a propagation model which does not accurately represent radio propagation can result in large distance estimation errors there by position estimation. Radio propagation modelling is a widely researched field for various wireless services and environments using different frequencies. Radio propagation modelling for indoors can be particularly difficult as the radio wave propagation in indoor environment depends on building structure, furniture in the building and the building materials used.

Radio propagation models are widely divided into two categories, Deterministic models and empirical models [53]. Deterministic models use the laws governing radio wave propagation to determine RSS at a particular point in an environment [54]. These models require large amount of details about the environment such as the 3-dimentional geometry details. One of the major techniques used in deterministic models is ray tracing. Some of the research using ray tracing techniques for propagation path loss modelling is detailed in [55] to [58]. Deterministic models are therefore accurate but site specific. In case of empirical models, the environment is included in the model as a series of random variables and does not depend on specific details about the environment such as number walls. These models are easy to build but are not as accurate.

Most popular and widely used propagation models are the empirical models. These models predict path loss as a function of parameters such as distance between transmitter and receiver. Measurements are collected throughout the environment and statistical methods are used to analyse this data to build a model. There has been extensive research done into the cellular mobile radio signal propagation and some of the research methods are described in [59] to [63]. The research involved realizing path loss models for cellular signal in urban areas with high rises to estimate the signal coverage. Some of the popular models in this area are Okumura – Hata model [59] [64] and Cost 231 – Hata model [61].

Indoor environment is quite different to outdoor environments and hence using models and parameters realised for outdoor environment, in an indoor environment would result in errors. Realizing a path loss model for indoors is complicated as the environment is dependent on the layout of the building and materials used. Moreover, the environment can change depending on factors such as, number of people in the building and if the doors and windows are open or closed. Propagation models for cellular mobile radio signals have been developed in indoor models. Some of these are Cost231 multi wall model [65] and Ericsson model [66] [75]. These models were realized for 900MHz to 1800MHz which covers the cellular radio frequency. A comparison of propagation path loss models for 2.5GHz frequency range is provided in [67]. There are two path loss models which are widely used for Wi-Fi radio signals in indoor environments.

1) The log-distance path loss model
2) The ITU (International Telecommunication Union) indoor path loss model

2.7.1 Log distance path loss model

Log - distance path loss model does not need site specific information such as the number of walls or floors between the transmitter and receiver. The model is given by

$$L_{Total} = PL(d_0) + 10\gamma \log_{10}(\frac{d}{d_0}) + X_s \quad \ldots\ldots. (2.17)$$

Where L_{Total} is total path loss

$PL(d_0)$ is path loss at reference distance d_0, but usually taken as (theoretical) free space path loss at 1m

‘ γ ’ is path loss co-efficient

‘d’ is the distance between transmitter and receiver

‘d_0’ is reference distance, but usually taken as 1m

X_s is a Guassian random variable with zero mean

2.7.2 The ITU indoor path loss model

ITU indoor path loss model can be considered as a modification of the Wall and Floor factor model. One approach proposed in [63] was to characterise indoor path loss with a fixed path loss co-efficient of ‘2’, which is same as free space, and then add the loss factors corresponding to walls and floors. The Wall and floor model can mathematically be represented as follows

$$L_{Total} = L_1 + 20\log r + n_f a_f + n_w a_w \quad \ldots\ldots. (2.18)$$

Where L_{total} is the total path loss

L1 is the loss at 1m distance

‘r’ is the distance between transmitter and receiver

‘n_f’ is number of floors

‘n_w’ is number of walls

‘a_f’ is attenuation factor per floor

‘a_w’ is attenuation factor per wall

It can be noticed from equation (2.18) that the site-specific information such as number of walls and floors is needed to realise this model. There were no values for ‘a_f’ and ‘a_w’ mentioned in [63].

The ITU model is similar to the model described above except that only losses due to the floors are considered explicitly. The path loss at points on same floor is calculated by altering the path loss co-efficient [62]. The model can be represented as follows

$$L_{Total} = 20\log f_c + 10\gamma \log r + n_f a_f - 28 \quad \ldots\ldots.. (2.19)$$

Where L_{total} is the total path loss

f_c is the frequency of the radio wave

‘ γ ’ is path loss co-efficient

‘n_f’ is number of floors

‘a_f’ is attenuation factor per floor

‘r’ is the distance between transmitter and receiver

Instead of a fixed path loss co-efficient as in the wall and floor factor model, the ITU model accounts for the loss between two points on the same floor by changing the path loss co-efficient (γ). The recommended path loss co-efficient for the model to be used in various environments are summarised in table 2.1 [62].

Frequency	Residential	Office	Commercial
900MHz	-	3.3	2.0
1.2 – 1.3 GHz	-	3.2	2.2
1.8 – 2.0 GHz	2.8	3.0	2.2
4 GHz	-	2.8	2.2
60 GHz	-	2.2	1.7

Table 2.1: Path loss co-efficient (N) values for ITU model

2.7.3 Probabilistic Signal propagation modelling

Wi-Fi radio signal modelling indoors is challenging as discussed in the previous sections. Instead of relying on the environmental features such as walls and floors to build a propagation model, probabilities for the measurements are calculated based on the calibration data. Some of the statistics for probability estimation are Histograms and Gaussians as discussed earlier. However, one of the limitations is that these probabilities could not be generated for locations with no calibration data. One of the solutions proposed in [27] is use of Gaussian Processes to build a radio map and estimating location of mobile device based on the radio map. Gaussian process estimates posterior distribution over the functions from calibration data. These distributions are represented non-parametrically based on the calibration data. The covariance between the calibration data is defined by a kernel function and the most widely used kernel is the Gaussian kernel. In [27] and [172] the radio map is used to predict the location given a position request RSS scan. These radio maps are helpful especially in reducing the calibration effort [169] and

updating the calibration data [170]. Both [170] and [171] claim the computational complexity to be the order of $O(n^3)$ where n is the number of calibration points. To keep the computational requirements of current system low, this regression technique is not implemented in the current work.

2.8 Conclusion

The previous sections describe various positioning techniques using radio signals. There are two widely used techniques which are lateration and fingerprinting. There are drawbacks associated with both techniques. There are various lateration techniques discussed in section 2.2.3 of which the most widely used is the one that estimates distance to the beacons. Section 2.4 describes various techniques to estimate the distances to beacons required for lateration techniques. The most popular distance estimation technique is to use the RSS, which requires a robust and effective path loss model. Some of the propagation models for Wi-Fi radio waves were described in section 2.7. There is no single path loss model which can be used for all indoor environments. Another requirement of lateration techniques is that the exact location of the Wi-Fi beacons should be known. This information can usually be acquired from the people who layout Wi-Fi infrastructure for a building. But if the positioning system is to be scaled to cover wider area such as a city, collecting Wi-Fi location information is not always feasible in the current state of the art.

The alternative to lateration techniques which have been widely used are the fingerprinting techniques. Some of these techniques were discussed in section 2.3. All the fingerprinting techniques require generating a database which is used during positioning phase. The size of the database hugely increases as these techniques are scaled up such as to a city level. Moreover, the signal comparison processes described in section 2.3 are computationally intensive and the size of database increases the computational intensity increases as well.

The proposed novel hybrid positioning system addresses the drawbacks of both positioning techniques. The system uses both fingerprinting and lateration techniques and reduces size of database, there by the computational intensity. The system also proposes a lateration algorithm with geographical coordinates which makes the system scalable.

Chapter 3

Hybrid System architecture and Implementation

3.1 Introduction:

Chapter 2 describes various positioning systems and techniques used in those systems. The most widely used techniques are lateration and fingerprinting which have some drawbacks as discussed in chapter 2. This chapter describes a novel hybrid positioning system which combines lateration and fingerprinting techniques to addresses the drawbacks of individual positioning techniques. The proposed system also has a calibration phase and positioning phase. The architecture of the system is discussed in the next section and also details differences to the usual fingerprinting and lateration techniques. One of the main aims of the system is to reduce size of database that needs to be stored. This is achieved by processing the data collected during calibration phase before storing it to a database. The system also uses a lateration technique which requires information regarding the exact location of each Wi-Fi beacon. In this initial proposed system, the Wi-Fi beacon location information is being assumed to be available. [119] published recently, describes a similar system. [119] divides the test area into blocks of approximately 6m×6m and calibration data is collected inside each block. Data from each block is used to calculate coefficients of a curve that fits the calibration data for each block (as the paper calls it subarea). These coefficients along with all the calibration data is stored. During positioning phase, firstly a Nearest neighbour algorithm is used to estimate the block and then uses the coefficients from curve fitting to refine the position

within the block. The system in [119] reported an accuracy of up to 3m. The system proposed in current work, uses the block propagation models to first identify the block and then uses the same models to refine the position within the block.

3.2 System architecture

The system has two phases, as in a usual fingerprinting positioning system, calibration phase and online phase. The following sections describe the overall system architecture in detail. The calibration phase involves dividing the area of interest into blocks. Calibration points are selected such that each block is covered. The data collected at the calibration points is processed to build a local propagation model for each block. These models are then stored to a database which can be used during positioning phase. Along with the propagation models, data from calibration point at centre of the block is also stored in database. During positioning phase, the RSS values in the position request are used to determine the block in which mobile device is located. After determining the block in which mobile device is located, propagation models stored in database from corresponding block are used to estimate distances to beacons. These distances are used to refine mobile device's position within the block using lateration techniques.

3.2.1 Calibration phase

During calibration phase, Wi-Fi beacon data is collected at multiple calibration points. To explain the architecture of proposed system, consider an area of interest covered by 'n' Wi-Fi beacons (AP_1, AP_2, AP_3 … AP_n) as shown in *figure 3.1*. The area of interest is divided into four blocks with multiple calibration points inside each block and each calibration point is represented by a star. Wi-Fi beacon data is collected at each calibration point.

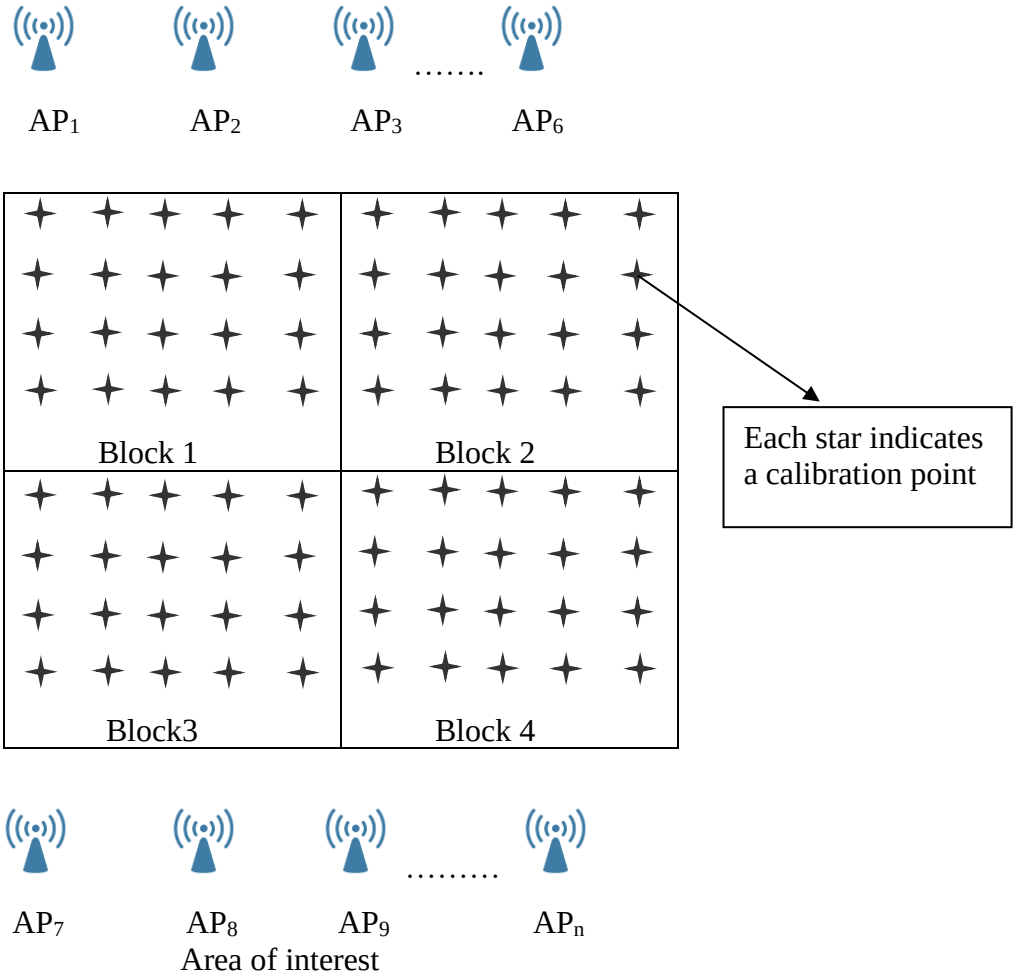


Fig. 3.1 Area of interest divided into multiple blocks showing calibration points

3.2.1.1 Data collection at calibration points

The Wi-Fi beacon data is collected at each calibration point by scanning for available Wi-Fi signals. The collected data contains RSS of each Wi-Fi beacon scanned at the calibration point. The scan data is tagged with the location of the calibration point. The data collected at a single calibration point in the area of interest described in previous section is detailed in figure 3.2.

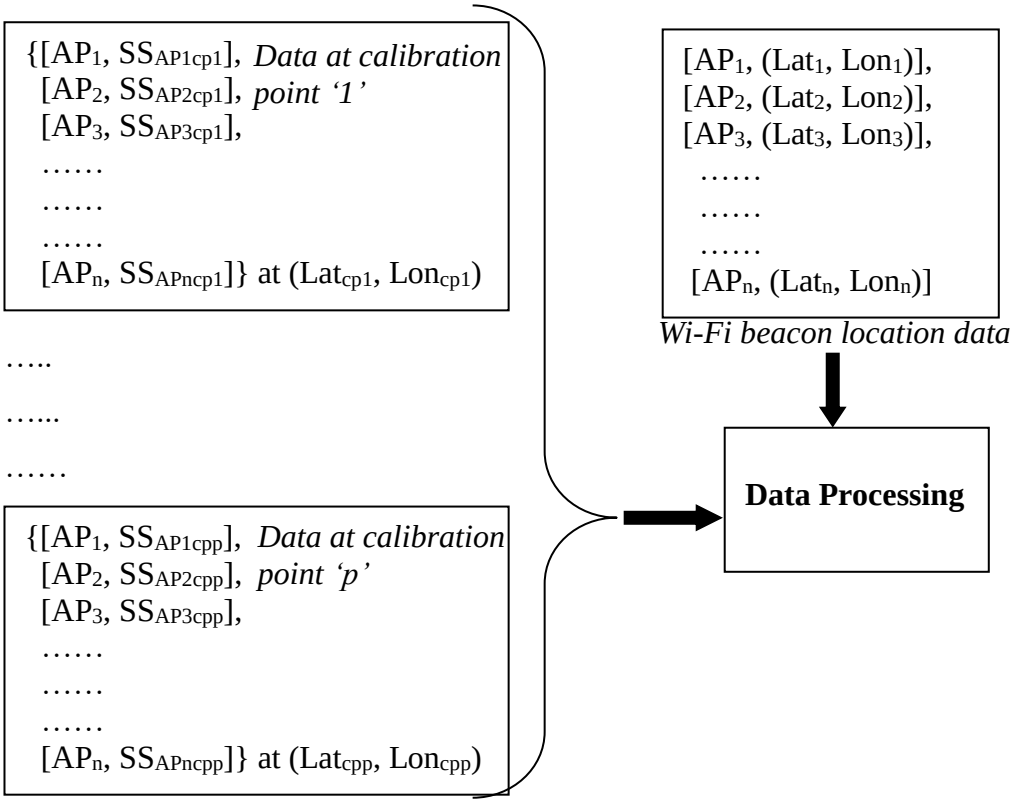
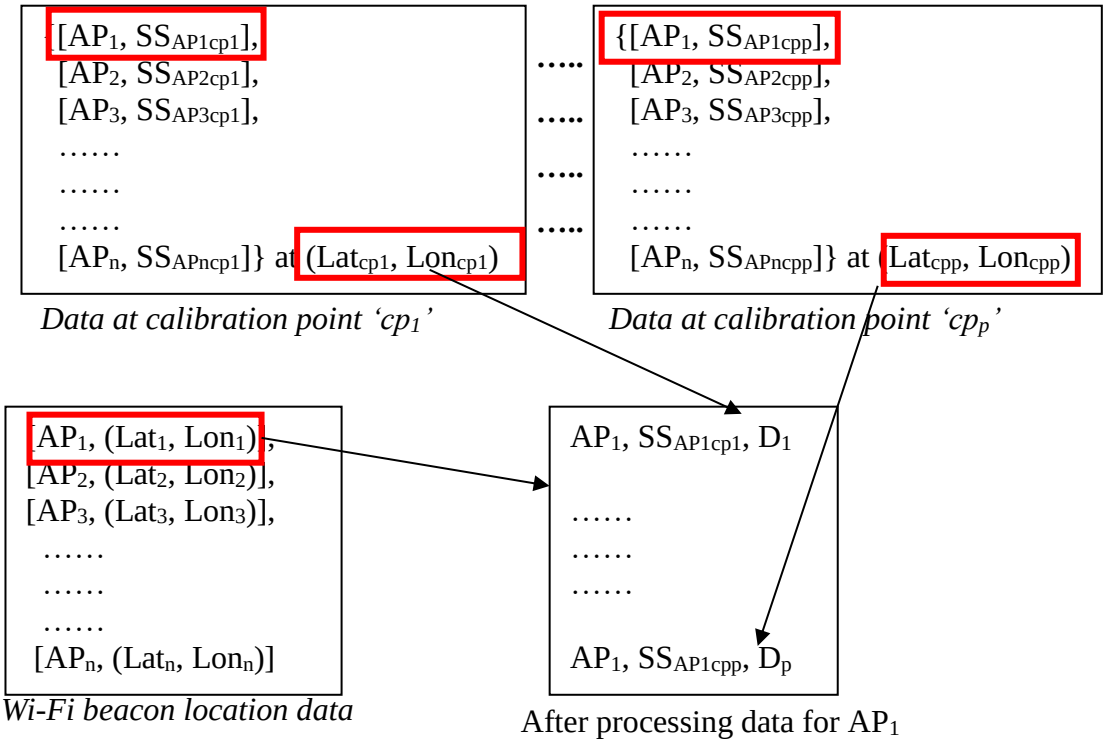


Figure 3.2 Data at each calibration point of a single block

Each scan contains MAC id to identify the Wi-Fi beacon and corresponding RSS at the calibration point. If each block is considered to have 'p' calibration points, there would be 'p' scans for the whole block as shown in figure 3.2. The system also assumes that location of each Wi-Fi beacon is known. Wi-Fi beacons location data consists of MAC ids of 'n' Wi-Fi beacons and their corresponding location which is also shown in *figure 3.2*. The scan data from 'p' calibration points and the Wi-Fi beacons location data is processed before being saved to a database.

3.2.1.2 Processing of calibration data

The data collected at calibration points within a single block is processed to generate a propagation model for each Wi-Fi beacon with in the block. Data collected is processed to get distance between Wi-Fi beacon and calibration points and corresponding RSS values. This processing is explained in figure 3.3 which illustrates data used for Wi-Fi beacon 'AP_1'.



D_1 is the distance between calibration point '1' and Wi-Fi beacon 'AP_1'

D_p is the distance between calibration point 'p' and Wi-Fi beacon 'AP_1'

Figure 3.3 Data processing for each Wi-Fi beacon in a block

The distances between calibration points and Wi-Fi beacons, D_1, D_2… D_p ('p' calibration points), are calculated based on the co-ordinate system used. Figure 3.3 shows coordinates of calibration points and Wi-Fi beacon location in geographical co-ordinate system which represents each point on earth with a latitude and longitude. The coordinates of calibration point 'p' are represented as (Lat_{cpp}, Lon_{cpp}) and coordinates of Wi-Fi beacon 'n' are (Lat_n, Lon_n). The distance (D_p) between the calibration point and Wi-Fi beacon is obtained by calculating the distance between two points (Lat_{cpp}, Lon_{cpp}) and (Lat_n, Lon_n). The distance (D_p) can be calculated using the following formula (also known as "Haversine formula" [68] shown in equations (3.1), (3.2) and (3.3).

$$D_p = R * C \qquad \ldots\ldots (3.1)$$

Where R is radius of earth, mean radius is 6371 Km

C is calculated using the following formula

$$C = 2 * a\tan 2(\sqrt{A}, \sqrt{(1-A)}) \quad \ldots\ldots (3.2)$$

A is calculated using the following formula

$$A = \sin^2(\frac{\Delta\varphi}{2}) + \cos\varphi_1 * \cos\varphi_2 * \sin^2(\frac{\Delta\lambda}{2}) \quad \ldots\ldots.. (3.3)$$

Where $\Delta\varphi$ is (Lat_{cpp} - Lat_n) in radians

$\Delta\lambda$ is (Lon_{cpp} - Lon_n) in radians

φ_1 is Lat_{cpp} in radians

φ_2 is Lat_n in radians

If a three-dimensional Cartesian co-ordinate system was used, the location coordinates can be represented as (X, Y, Z). The coordinates of calibration point 'p' are (X_p, Y_p, Z_p) and coordinates of Wi-Fi beacon 'n' can be represented as (X_n, Y_n, Z_n). The distance D_p can be calculated using Pythagorean theorem as follows shown in *equation (3.4)*.

$$D_p = \sqrt{(X_p - X_n)^2 + (Y_p - Y_n)^2 + (Z_p - Z_n)^2} \quad \ldots\ldots (3.4)$$

3.2.1.3 Processing data from a single block

Processing of data collected at all calibration points within a block is carried out for each Wi-Fi beacon. Data processing for each Wi-Fi beacon is illustrated in figure 3.4. Following initial processing, data has distance and RSS information for each Wi-Fi beacon, linear regression technique described in [69] [70] is used to model the relationship between distance and RSS. Log-distance propagation model discussed in chapter 2 shows that RSS (dB) varies linearly with log-distance. Linear regression of RSS (dB) and log-distance provides a propagation model [71] for each Wi-Fi beacon, which is specific to that block. Figure 3.4 shows the models represented by equations. The equation $f_{AP1}(SS, D)$ represents a propagation model for Wi-Fi beacon AP_1 with in block B_1. The coordinates of the centre of the block (Lat_{B1}, Lon_{B1}) are attached to the list of propagation models realised for each Wi-Fi beacon with in the block. This process is repeated for all blocks in the area of interest.

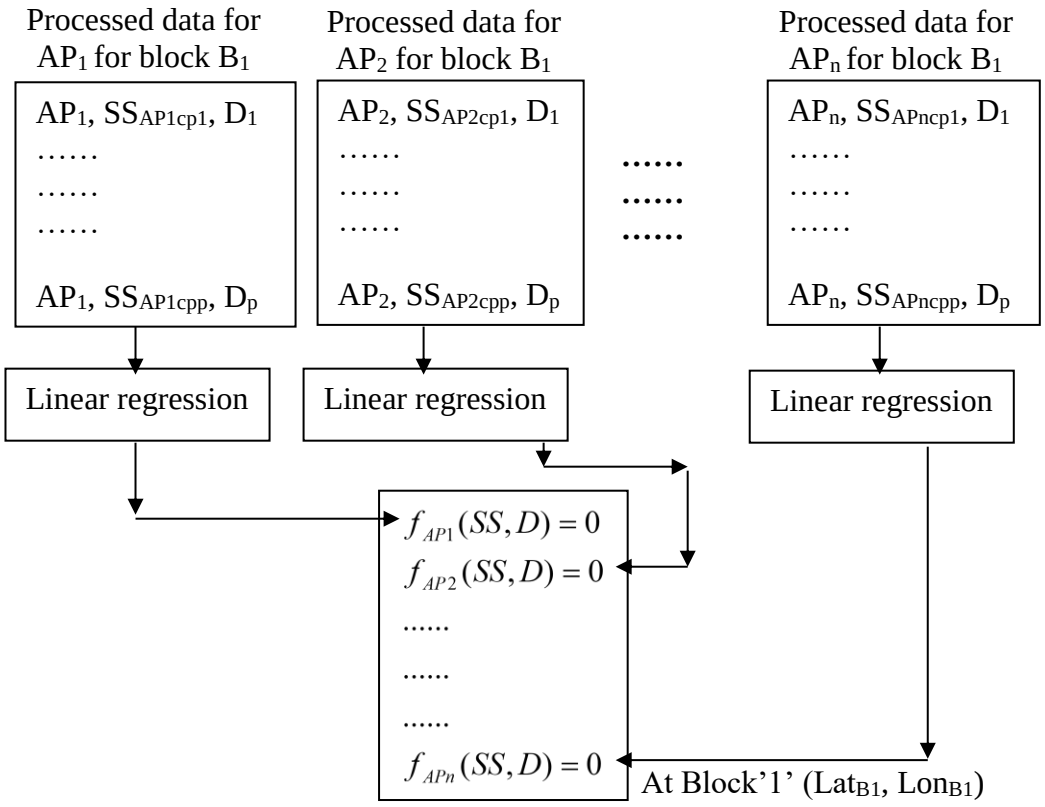


Figure 3.4 Data processing for a single block

3.2.1.4 Database for positioning system

Figure 3.5 shows an overview of data stored to the database. The propagation models of each Wi-Fi beacon in each block are shown. For example, propagation model denoted as $f_{AP1}(SS,D)=0$ for Wi-Fi beacon AP_1 belongs to block '1'whose centre is at (Lat_{B1}, Lon_{B1}). Block '1' has a list of 'n' propagation models for each Wi-Fi beacon. Along with this data, calibration data the centre of each block is also saved to the database. Furthermore, location data of all Wi-Fi beacons is also stored as part of the database.

In a usual finger printing system, the scan data from all calibration points is stored to a database. Considering the current example, the area of interest has 'q' blocks and 'p' calibration points in each block. The number of scan data arrays from all the calibration points that need to be stored to the database would be $p \times q$. The current proposed system only stores 'q' scan data arrays, one from the centre of each

block. This results in a huge reduction of the amount of data that needs to be stored to database.

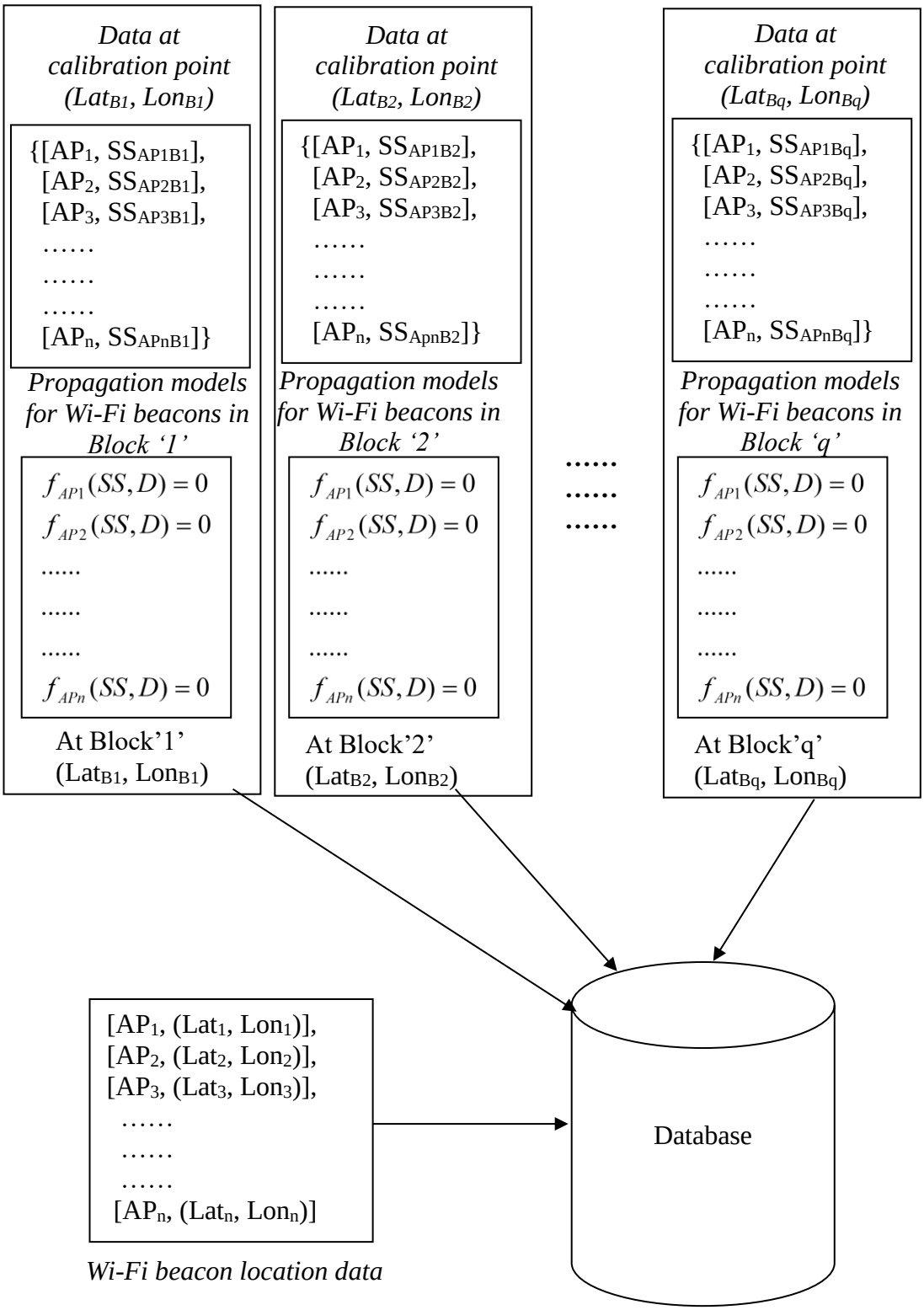


Figure 3.5 Database for an Area of interest with 'q' blocks

3.2.2 Positioning phase

This section describes working of the proposed system during positioning phase. During positioning phase, the mobile device scans for the available Wi-Fi beacons and generates a position request. Position request contains a list of Wi-Fi beacons scanned at the unknown location and corresponding RSS detected. Consider a scenario illustrated in *figure 3.6* where a mobile device is at an unknown location and can scan 'n' Wi-Fi beacons and the corresponding RSS are indicated as $\overline{SS}$.

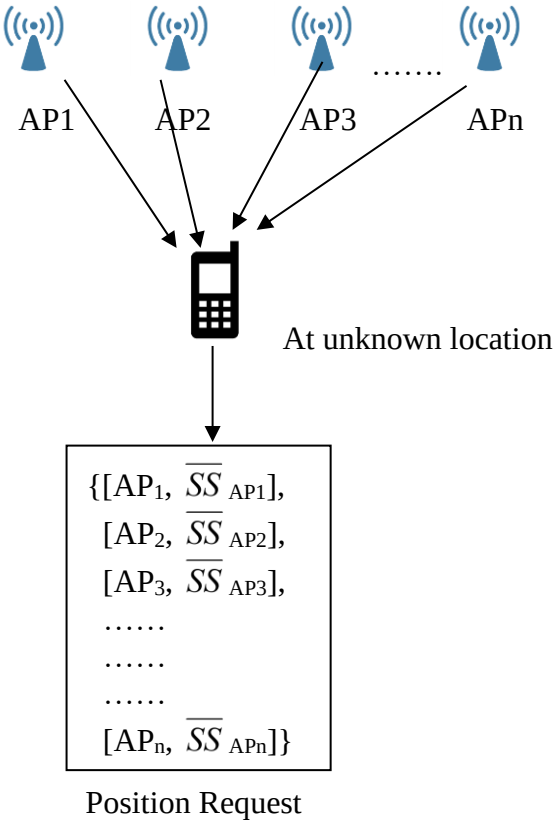


Figure 3.6 Position request

The data in position request is first used to identify the block in which the mobile device is located and then a refined location is calculated using propagation models corresponding to the identified block.

3.2.2.1 Block identification

The database generated during calibration phase has scan data collected at the centre of each block. These scans are matched with the position request as described

in section 2.3.2. The signal distances (also referred as Euclidean distance [72]) to the scans from each block are calculated and the scan with least signal distance is determined. The scan resulting in smallest signal distance determines the block in which mobile device is located.

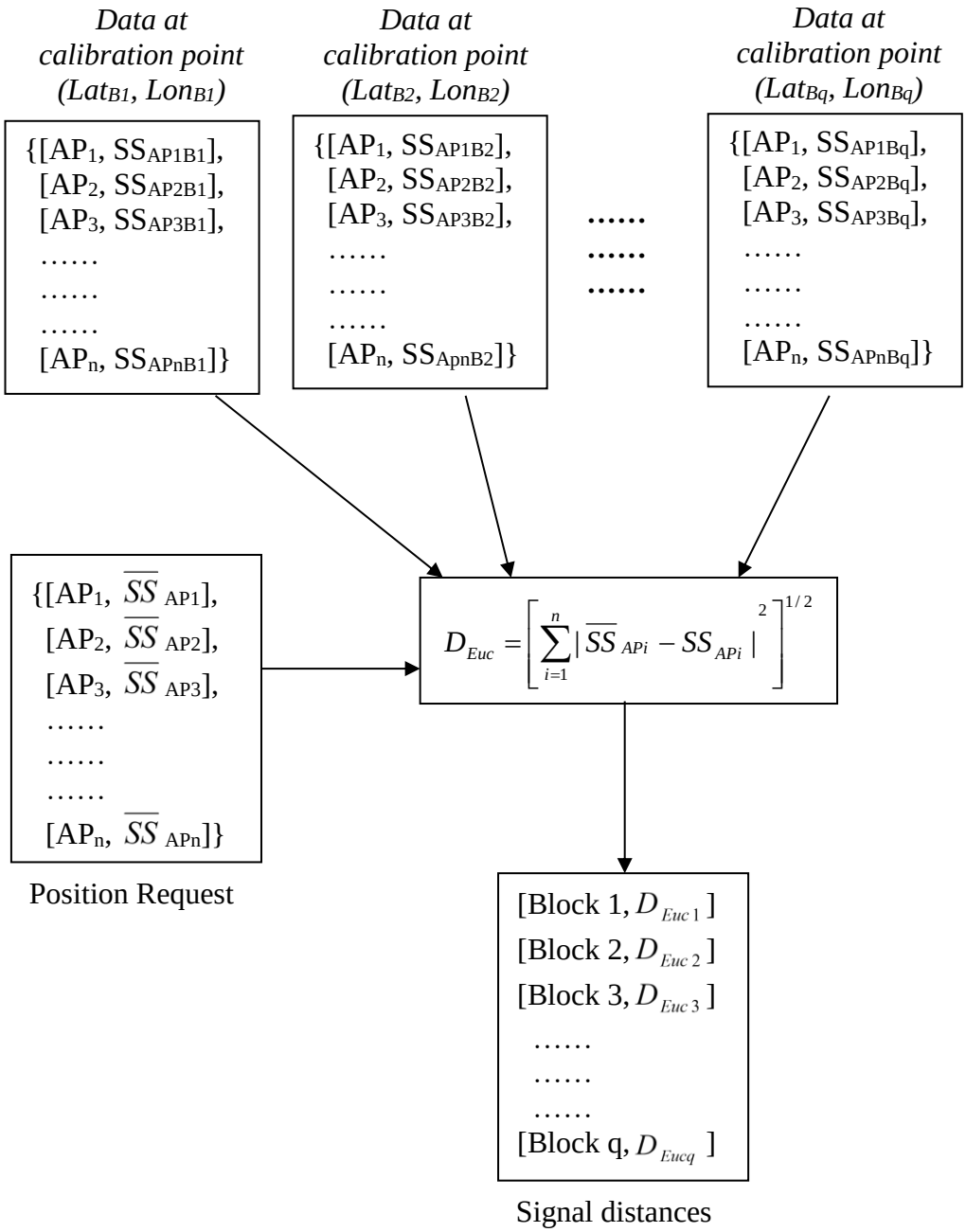


Figure 3.7 Signal distance calculation

Figure 3.7 illustrates calibration data from 'q' blocks and a position request. The calculation of Euclidean distances (D_{Euc1}, D_{Euc2}, D_{Euc3}, ….. D_{Eucq}) between

position request scan and calibration scans stored on the database is also illustrated in figure 3.7. The smallest distance determines the block in which the mobile device is located.

3.2.2.2 Distance estimation

The previous section describes estimation of the block in which the mobile device is located within the area of interest. The position can be refined further by lateration, which requires estimating distances to Wi-Fi beacons. The distances can be estimated using corresponding propagation models for Wi-Fi beacons in identified block. *Figure 3.8* illustrates the data used in distance estimation. Assuming Block 'i' was identified to be the mobile device's location, *Figure 3.8* shows the path loss models for Wi-Fi beacons seen in that block. Substituting the RSS values from the position request in the path loss models corresponding to block 'i' gives distances (D_{AP}) to the Wi-Fi beacons.

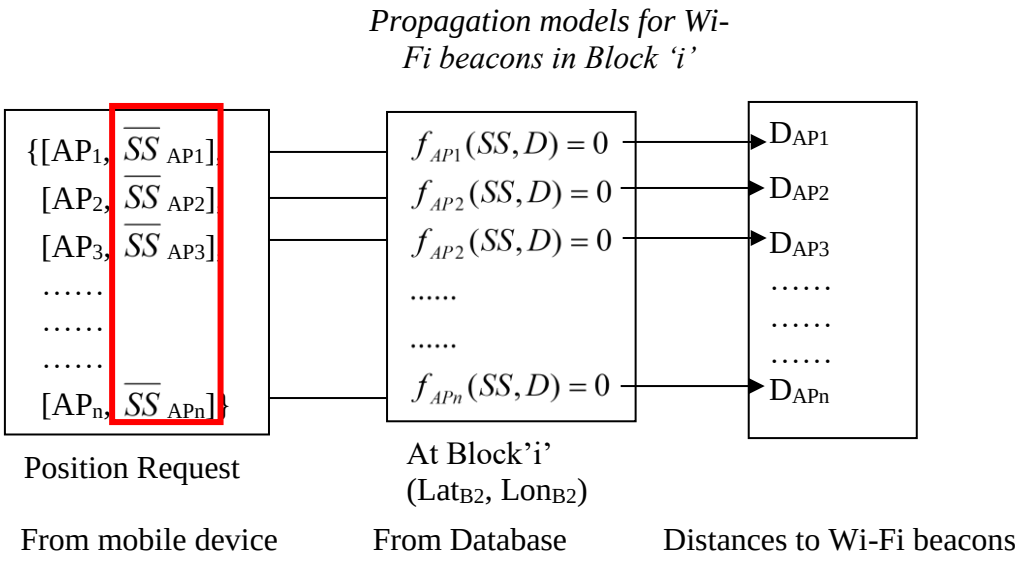


Figure 3.8 Distance estimation

3.2.2.3 Lateration

The estimated distances to Wi-Fi beacons are used to refine the mobile device's location within the block. Lateration using geographical coordinates require conversion to Cartesian coordinates. For the initial experimental implementation, a three-dimensional Cartesian coordinates system is considered. An algorithm to

convert geographical to Cartesian coordinates is proposed in the later sections. The following sections describe the experimental implementation of the proposed system.

3.3 Experimental implementation

The experimental implementation of proposed system is described in this section. The test area/area of interest is Sanderson building, King's Buildings campus of University of Edinburgh. This is a two-storey building and test areas considered are corridors on Ground (F1) and second (F2) floors. The floor plans for the building were obtained from the university. Figure 3.9 shows test area of the building on Google maps.

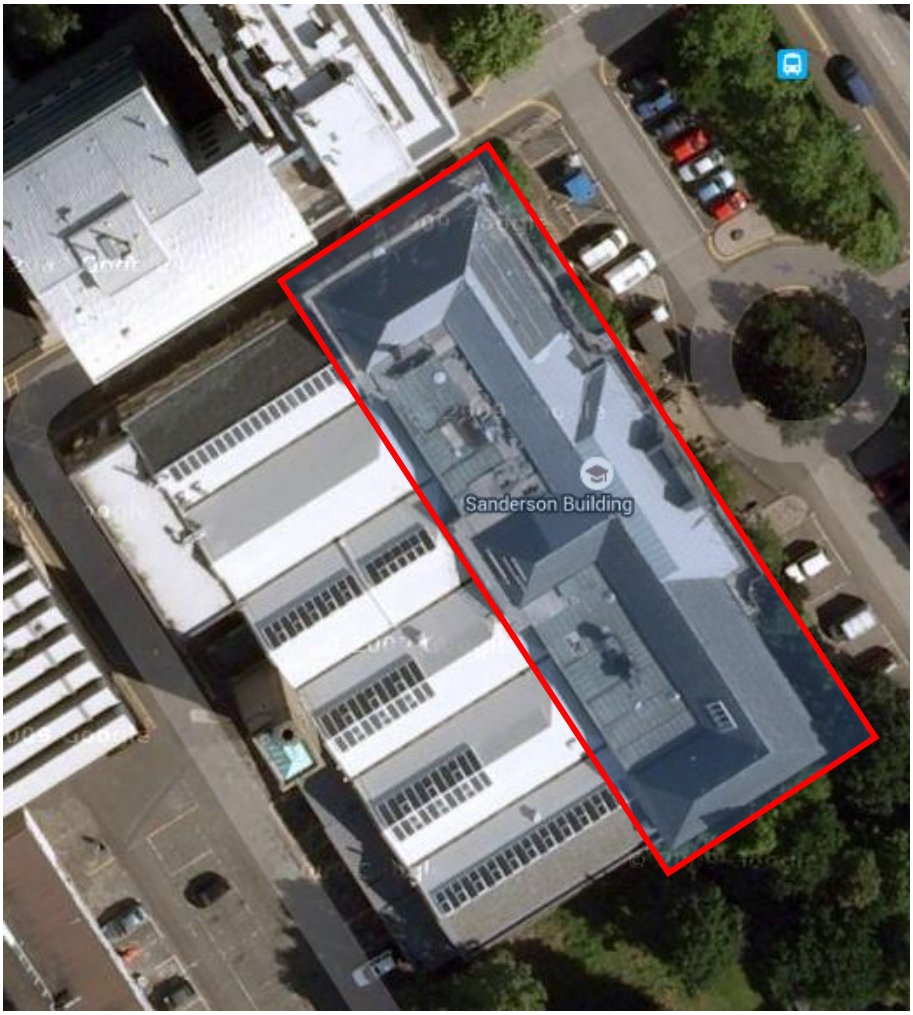


Figure 3.9 Sanderson building on Google maps

3.3.1 Calibration phase

The first task was to lay down a co-ordinate system for test area. The building is 17m wide and 60m long and Figure 3.10 shows the indoor floor plan of ground floor. The north-east corner of the building was considered to be the origin and orientation of X and Y axis are shown in figure 3.10. The accessible area on ground floor (F1) is highlighted in blue. The experiment relies on the existing Wi-Fi infrastructure instead of installing additional beacons. The Wi-Fi beacons were identified by physically surveying the area. There were three Wi-Fi beacons identified on the ground floor (F1) and two on the upper floor (F2). The location of these beacons is shown on the floor plan in figures *3.10 and 3.11*. The x-y coordinates of Wi-Fi beacons were determined using the floor plans.

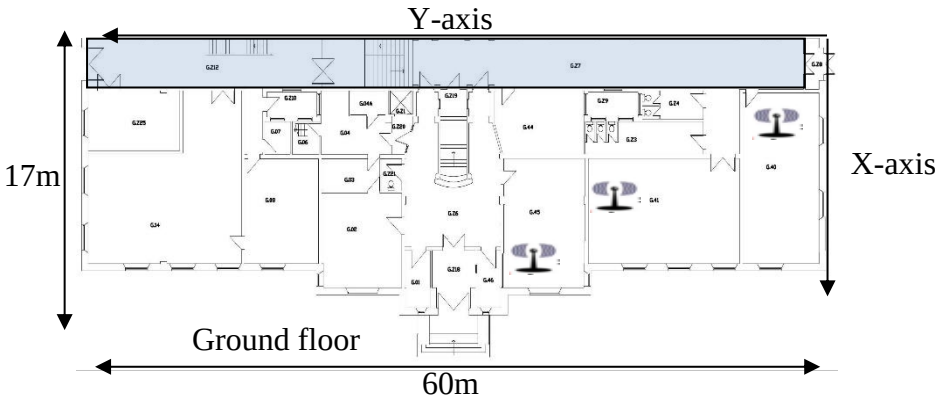


Figure 3.10 Ground floor (F1) with x-y axis and Wi-Fi beacons

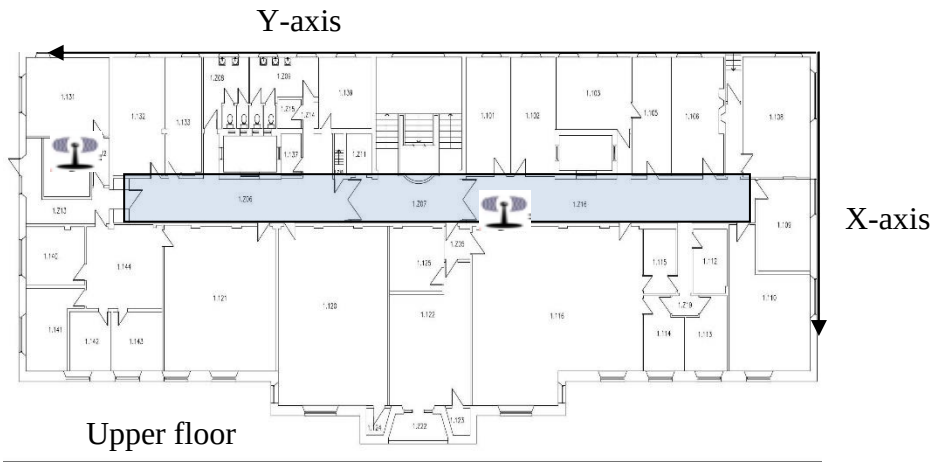


Figure 3.11 Upper floor (F2) with x-y axis and Wi-Fi beacons

3.3.1.1 Wi-Fi beacon location data collection:

During the survey to identify access points in the building, location of Wi-Fi beacons in corridors and accessible classrooms were identified. The height at which beacons were installed was also measured. Then using the floor plans, X-Y coordinates of beacons were calculated. Coordinates of the beacons can then be expressed in (X, Y, Z) format, where X, Y are the distances from respective axes and Z, the height from ground at which the beacon was installed. The Wi-Fi beacon location data is stored as [AP_i, X_i, Y_i, Z_i] where 'AP_i' is the MAC id of the Wi-Fi beacons. The location data for the five Wi-Fi beacons in test area is shown in table 3.1 and the distances are measured in meters.

MAC id	X (m)	Y(m)	Z(m)
00:13:5F:F9:23:D0	8.91	6.73	2.65
00:15:C7:A9:E6:01	12.13	19.25	2.97
00:12:44:B3:31:70	15.45	26.11	2.95
00:13:5F:F8:F3:F0	8.76	26	5.82
00:1B:8F:88:9C:20	4.46	56.15	5.82

Table 3.1 Wi-Fi beacon location data

3.3.1.2 Calibration points

The test area on ground floor (F1) and upper floor (F2) are highlighted in *figures 3.10 and 3.11* respectively. The test area on ground floor (F1) is 60m × 2.5m and Upper floor (F2) it is 40m × 2.5m. The test area on ground floor (F1) is divided into 12 blocks each of the size 5m × 2.5m and test area on upper floor (F2) is divided into 8 blocks of 5m × 2.5m. Figures 3.12 and 3.13 illustrate the test area on F1 and F2 divided into blocks.

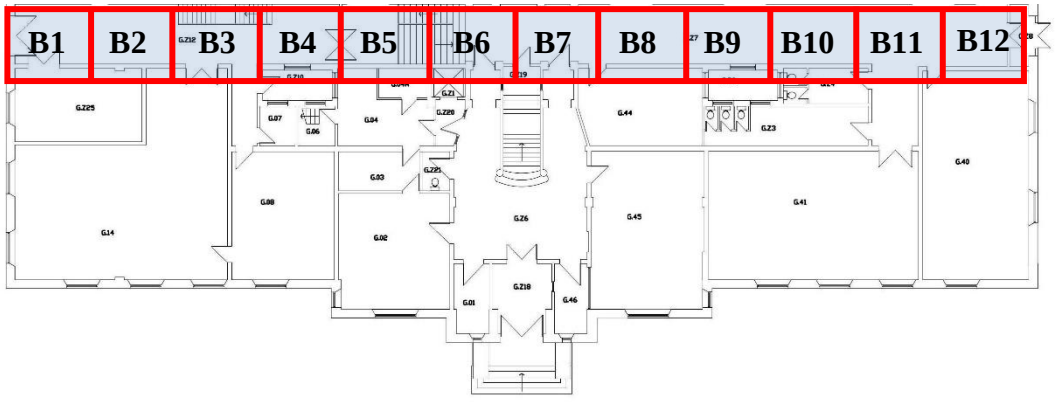


Figure 3.12 F1 test area divided into 12 blocks

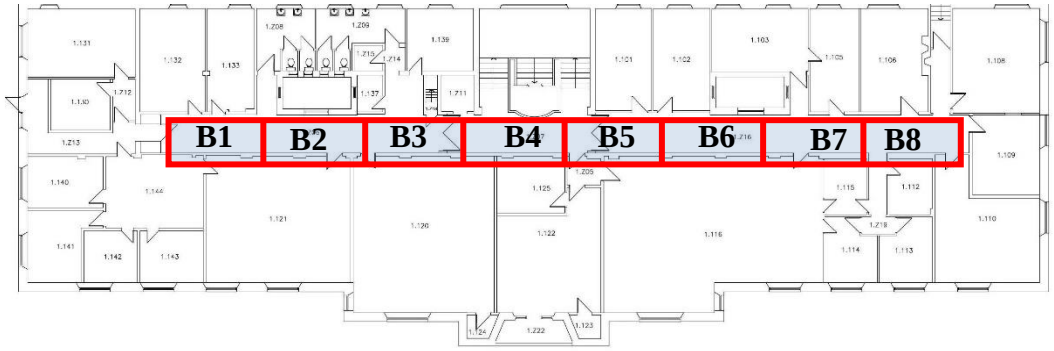
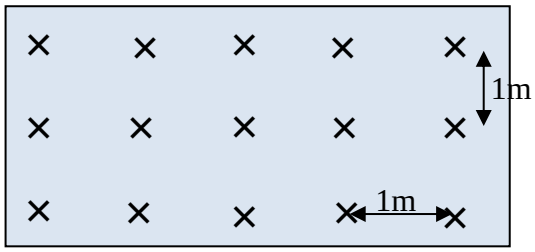


Figure 3.13 F2 test area divided into 8 blocks

Each block in the test area has 15 calibration points approximately 1m apart, which are shown in figure 3.14. However, block '12' on first floor is an exception which has nine calibration points. The test area has a total of 294 calibration points.



'×' – Represents a calibration point

Figure 3.14 Calibration points in a block

3.3.1.3 Data collection at calibration points

The data collection was done using a laptop running an application called "inSSIDer" [73]. The application scans for available beacons and displays the information such as MAC id, SSID (service set identifier), RSS and channel being used by the Wi-Fi beacon. A screenshot of the application is shown in *figure 3.15*. The application also stores all data collected between start and stop of the scanning to a log file. Since RSS varies due to various factors discussed in chapter 2, data was collected at each calibration point for 1 min. The data was collected facing north, south, east and west directions for 15sec each. Data collection in multiple orientations helps to compensate for the user's body between laptop and the Wi-Fi beacon. The scanning application creates a log file with the scan data in XML format. A screenshot of the log file is shown in figure 3.16.

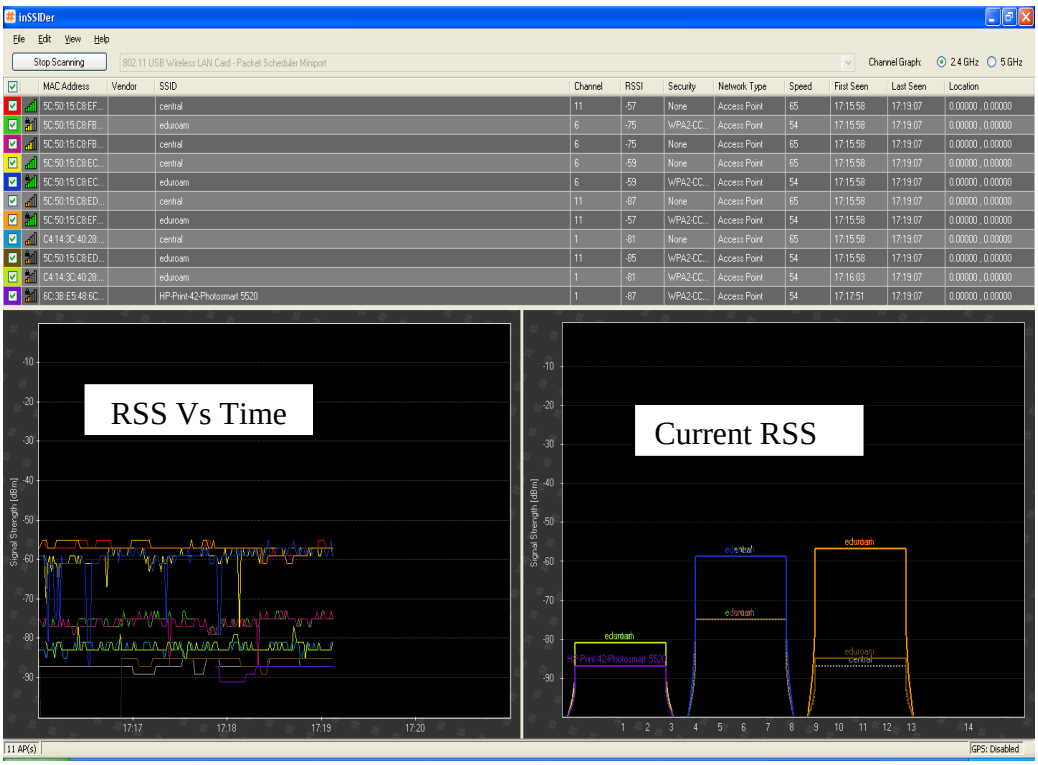


Fig.3.15 Screenshot of "inSSIDer" application

Figure 3.16 shows a part of the log file which shows the data stored for each Wi-Fi beacon scanned. The log file from application consists of data which is not needed for the positioning system. The data needed is MAC id and RSS which are highlighted in figure 3.16. This data has to be obtained from log file for all Wi-Fi beacons scanned during data collection process. The data collection process is repeated at all 294 calibration points in test area. The calibration process resulted in 294 log files which needed to be processed to extract the required information from them.

Figure 3.16 Screenshot of a section of the "inSSIDer" log file

3.3.1.4 Data processing

The data from log files was extracted with an application written in Java. This application goes through the log file and creates a new file which only contains the MAC id and average RSS information. The data from the log files is tagged with a location and format of the processed data is shown below.

$@CP_i\,(X_i, Y_i, Z_i)$

$[(AP_1,SS_{AP1}),\ldots..(AP_n,SS_{APn})]$ for i=1,2,..,P

Where AP indicates the MAC address of the Wi-Fi beacon,

SS_{AP} indicates the average RSS at calibration point

(X, Y, Z) indicate the coordinates of the calibration point

'n' is the number of Wi-Fi beacons scanned

'P' is the number of calibration points (15 per block in this case)

The processed data was stored in a text file and a screenshot of the data stored is shown in *figure 3.17*. The file contains coordinates of each calibration point, MAC id of the Wi-Fi beacons and corresponding average RSS values. The first line highlighted in *figure 3.17* shows X, Y and Z coordinates of calibration point. Following the calibration point coordinates, list of Wi-Fi beacons and corresponding RSS values are stored (also highlighted in the figure for one calibration point). RSS value stored for each Wi-Fi beacon is the average of RSS values measured over one minute at the calibration point during data collection.

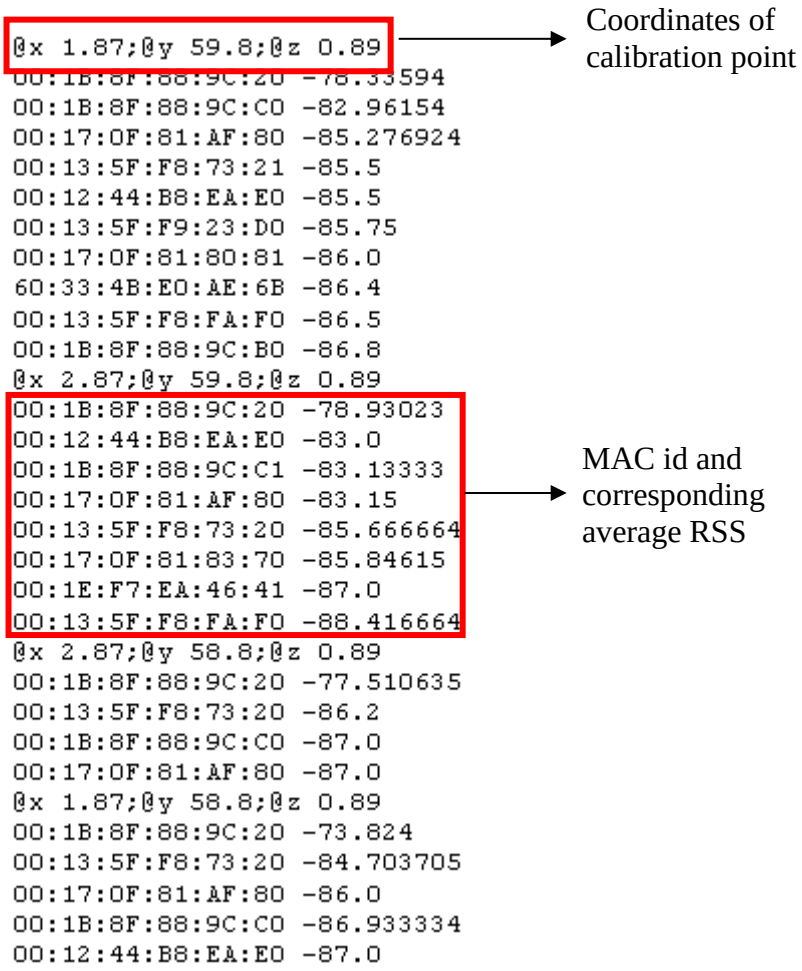


Figure 3.17 screenshot of data after the initial processing

In a conventional fingerprinting system, the data shown in *figure 3.17* is stored to a database. This data is used during positioning phase, to compare with position request. For proposed algorithm, the processed data is used to build propagation models for each Wi-Fi beacon within a block.

3.3.1.5 Generation of propagation models

The processed data along with the Wi-Fi beacon location data is used to generate propagation models. The process is illustrated using the data from block 4 on first floor as an example. A Java application was developed to calculate the distances between each Wi-Fi beacon and calibration points in a block. Figure 3.18 illustrates the distance calculation process for a Wi-Fi beacon with MAC id "00:13:5F:F8:F3:F0" at each calibration point in block '4' on first floor. The distances are calculated for all 15 calibration points with in block '4'. The distances between calibration points and corresponding RSS values for MAC id "00:13:5F:F8:F3:F0" in block '4' on first floor are illustrated in figure 3.18.

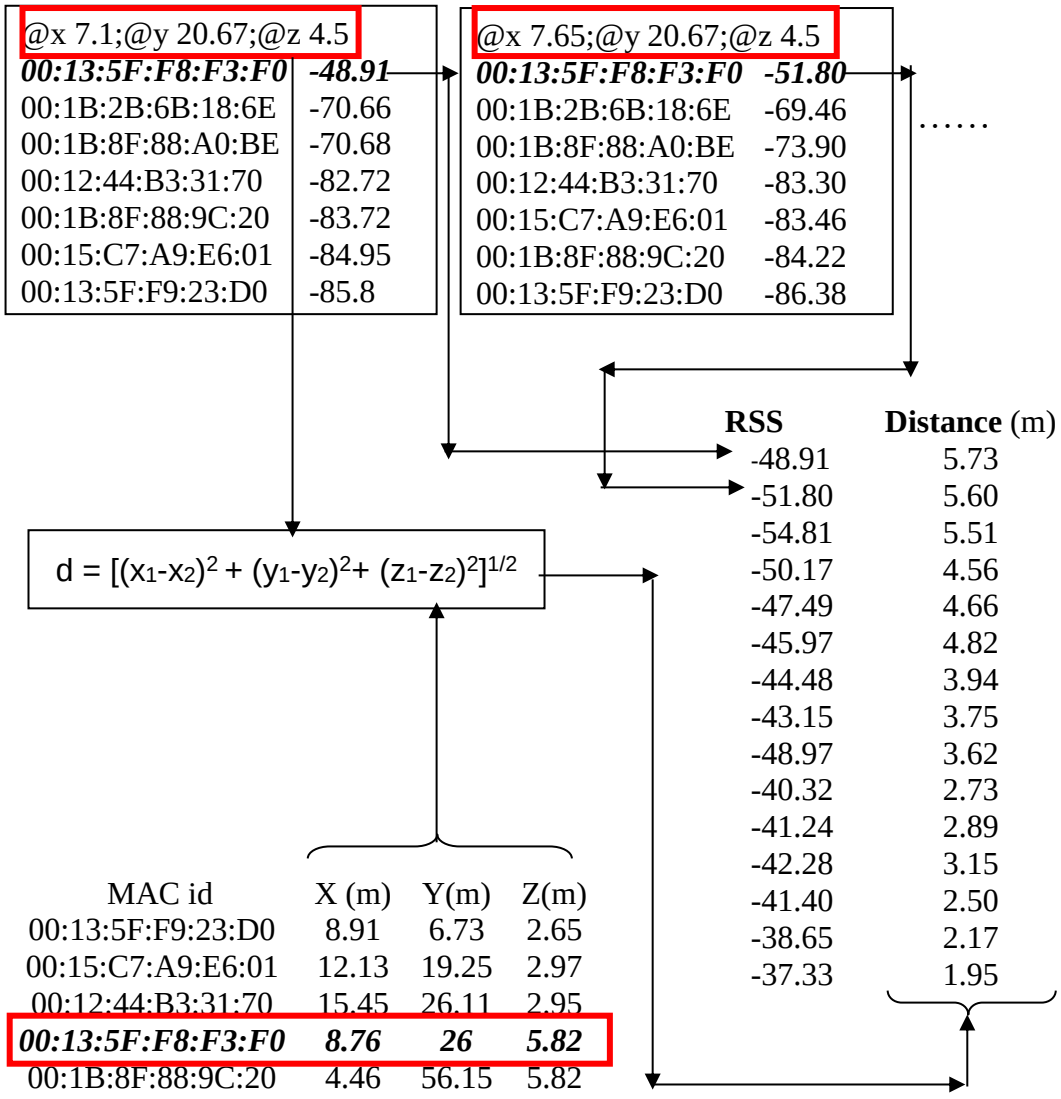


Figure 3.18 Distance calculation between Wi-Fi beacon and calibration points

The distances and RSS values from *figure 3.18* can be used to generate a propagation model for the Wi-Fi beacon "00:13:5F:F8:F3:F0" specific to block '4' on Upper floor (F2). The free space path loss model discussed in chapter 2, and equation 2.15 is the mathematical representation. Rewriting equation 2.15 [74] [75],

$$P_r(dB) = P_t(dB) - 40.04 - 20\log(R) \quad \ldots\ldots (3.5)$$

Where P_t is power at transmitter

P_r is power at the receiver

R is the distance between transmitter and receiver

Since the transmit power is fixed by manufacturer of the Wi-Fi beacon (Some Wi-Fi beacons can be programmed to change their power, but cannot be altered during operation) *equation (3.1)* can be rewritten as follows

$$P_r(dB) = P_{r0}(dB) - 10\gamma\log(R) \quad \ldots\ldots (3.6)$$

Equation (3.2) is similar to a straight-line equation,

$$y = mx + c \quad \ldots\ldots (3.7)$$

Comparing equations (3.6) and (3.7),

y = $P_r(dB)$ = Received power in dB

c = $P_{r0}(dB) = P_t(dB) - 40.04$

m = γ = propagation co-efficient

x = $10\log(R)$

Equations (3.7) and (3.7) show that linear regression modelling can be used on RSS and distance values to generate a propagation model. The RSS and distances calculated (as illustrated in figure 3.18) can be plotted and a linear line is fitted. The fitted line is the propagation model for the Wi-Fi beacon with in a particular block. *Figure 3.19* shows the plot of RSS and distance for Wi-Fi beacon "00:13:5F:F8:F3:F0".

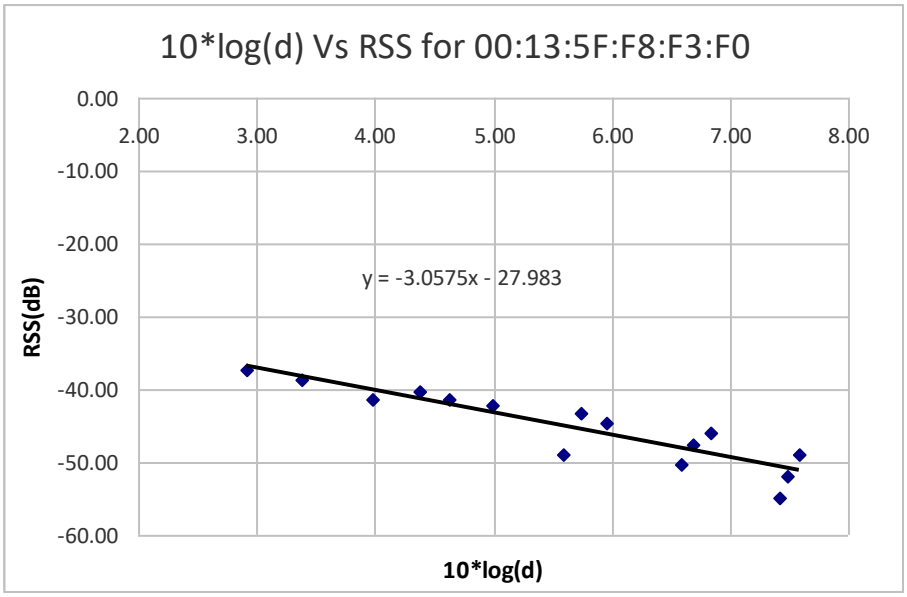


Figure 3.19 Plot of Distance Vs RSS for Wi-Fi beacon "00:13:5F:F8:F3:F0"

The propagation model for Wi-Fi beacon "00:13:5F:F8:F0" in block '4' is given by the equation

$$y = -3.0575x - 27.983 \quad \ldots\ldots (3.8)$$

In *equation (3.8)* 'y' is RSS in dB and 'x' is 10*log (d) where 'd' is the distance from Wi-Fi beacon. Path loss models are generated for all the other Wi-Fi beacons scanned in the block. There were five know Wi-Fi beacons in block'4' on first floor. *Figure 3.20* shows the plots for the five Wi-Fi beacons.

The Wi-Fi beacons and corresponding propagation models for block'4' are shown in *table 3.2*.

MAC Id	Propagation model
00:13:5F:F8:F3:F0	y = -3.0575x-27.983
00:15:C7:A9:E6:01	y = -0.7518x - 78.649
00:1B:8F:88:9C:20	y = 3.2376x - 133.12
00:12:44:B3:31:70	y = 0.9689x - 93.366
00:13:5F:F9:23:D0	y = 0.7227x - 95.17

Table 3.2 Propagation models for Wi-Fi beacons in block'4'

The data shown in table 3.2 is tagged with coordinates of centre of the block and stored to database. Along with propagation model data, the scan data collected at

calibration point at the centre of the block is also stored. The processing of data collected in the block' 4', leads to storing a single scan and propagation models for each access point to the database. This reduces the number of scans to be stored for block '4' from 15 scans to just one scan and some additional propagation model equations.

The process of fitting a line for the distance and RSS values is repeated for all the blocks and Wi-Fi beacons. The scan data from centre of each block along with the propagation models tagged with coordinates of centre of the block are stored as database to be used during positioning phase.

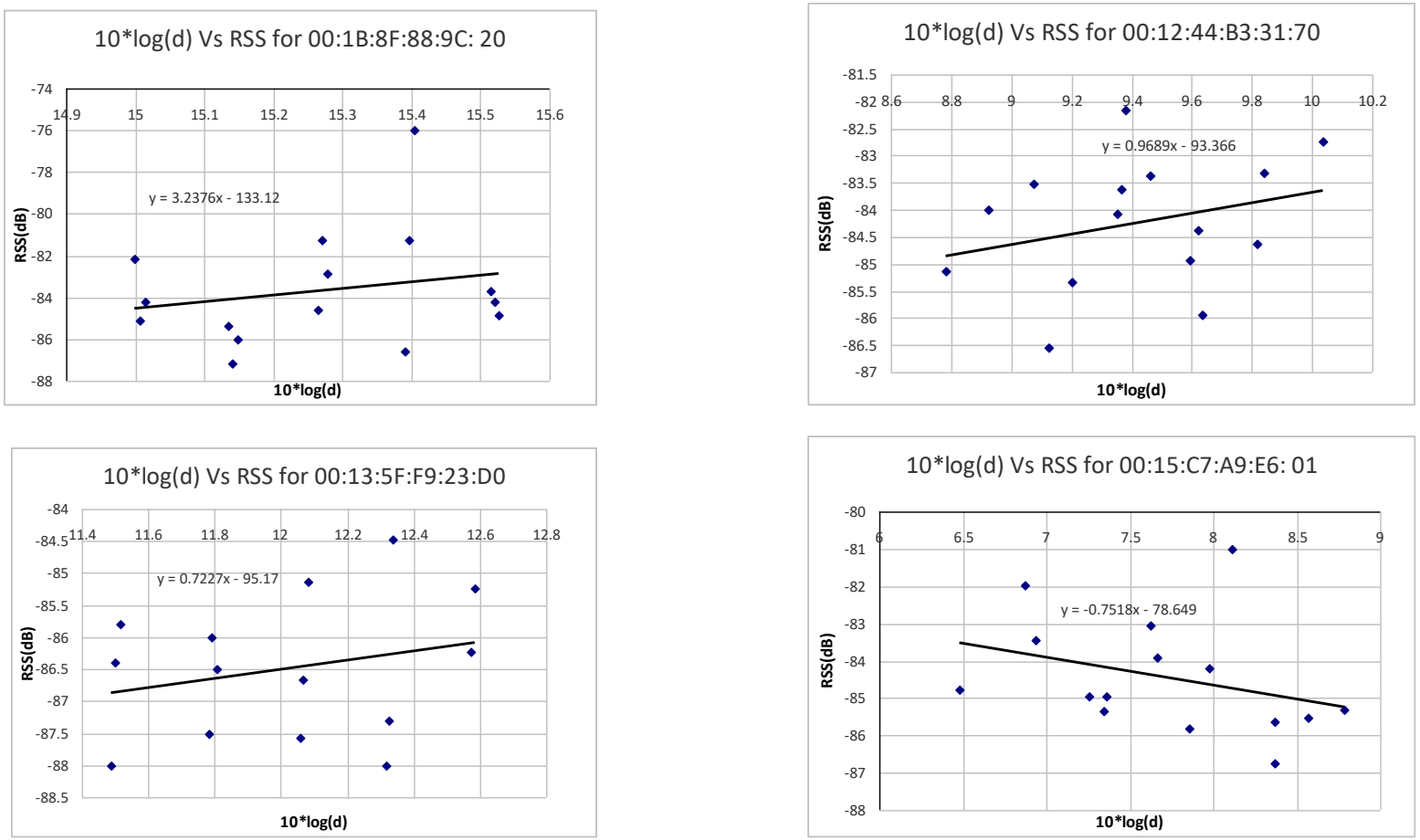


Figure 3.20 Plot of Distance Vs RSS for Wi-Fi beacons in block ‘4’

3.3.1.6 Database size comparison

The calibration scans at the centre of each block and the propagation models for all known Wi-Fi beacons in each block are stored to a database as a text file. The database for current proposed system is generated by processing data collected during calibration phase as described in previous sections. The size of hybrid system database text file is compared with fingerprint database size for each block in figure 3.21. The fingerprint database consists of calibration data collected at 294 calibration points where as the database for proposed system stores data from 20 calibration points.

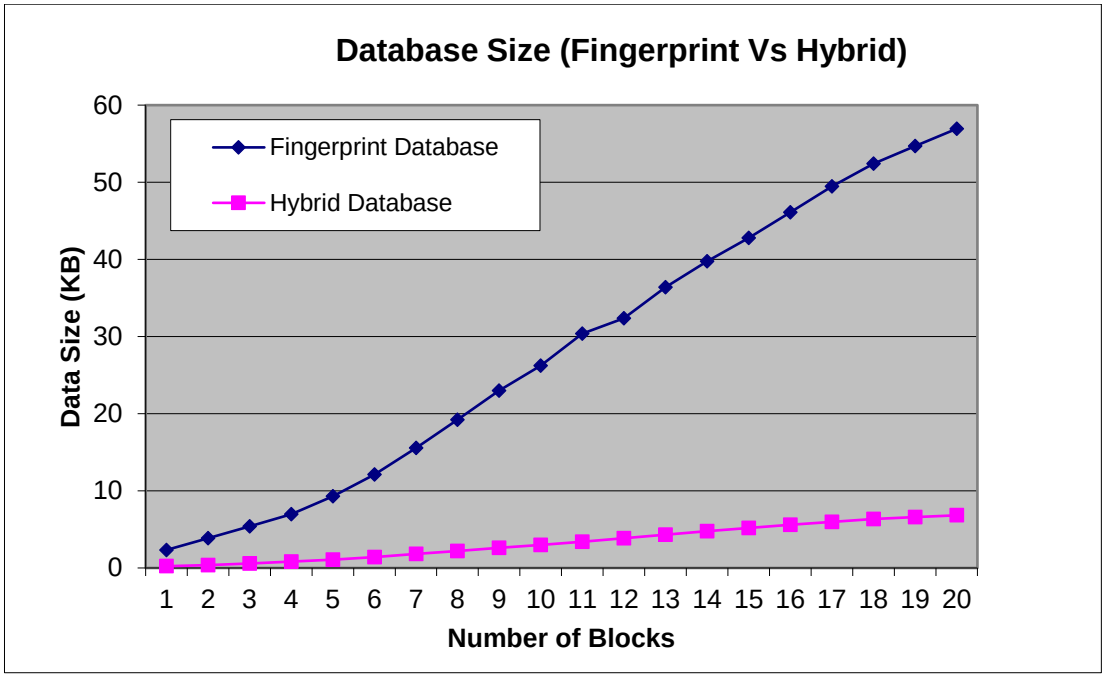


Figure 3.21 Comparison of Database size with Hybrid and Fingerprint systems

The size of hybrid database for current test area is 6.8KB whereas for a fingerprint system the database size is 56.9KB. The database size with proposed hybrid system is about eight times smaller compared to fingerprint system. A smaller database is particularly important for a scalable system, which is one of the main aims of current work.

3.3.2 Positioning phase

During positioning phase, position request data from the mobile device is processed along with the database generated during calibration phase by a positioning

algorithm to estimate location of mobile device. 20 test points were identified in the test area illustrated in figure 3.12 *and* 3.13. The locations of test points within the test area are shown in figure 3.22.

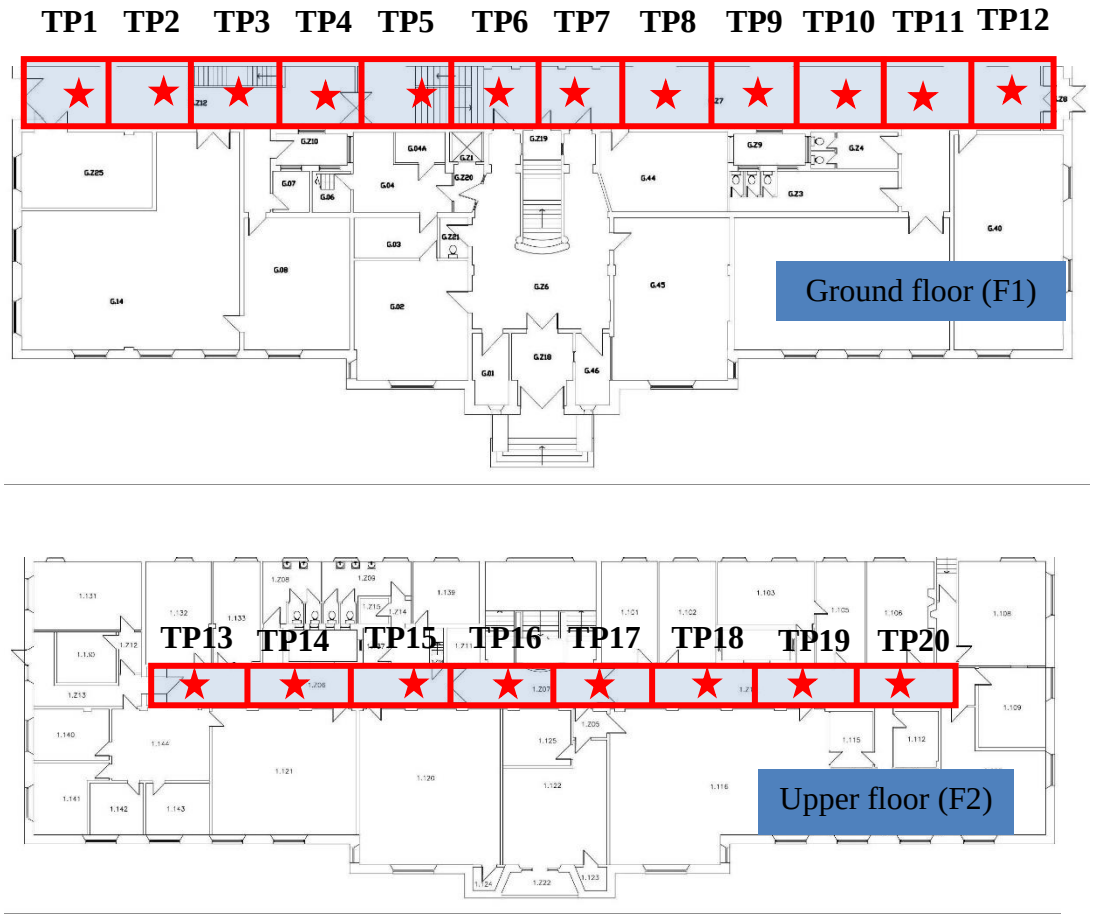


Figure 3.22 Location of test points

As indicated in *figure 3.22* one test point is associated with each block, ground floor has 12 test points and 8 test points in first floor. The following section details test point results which include accuracy of block identification, distance estimation to Wi-Fi beacons and lateration.

3.3.2.1 Block Identification results

The test points are spread over two floors and 20 blocks. The position requests are compared with the 20 calibration scans collected at centre of the blocks. Block identification for 12 test points on ground floor are as shown in table 3.3 and 8 test points on first floor in table 3.4. The test points for which a wrong block was

identified are highlighted. The accuracy of block identification over 20 test points is 80%.

Test Point	Actual block	Estimated block
TP1	B1	B2
TP2	B2	B2
TP3	B3	B4
TP4	B4	B3
TP5	B5	B5
TP6	B6	B6
TP7	B7	B8
TP8	B8	B8
TP9	B9	B9
TP10	B10	B10
TP11	B11	B11
TP12	B12	B12

Table 3.3 Bock identification results for test points on ground floor (F1)

Test Point	Actual block	Estimated block
TP13	B1	B1
TP14	B2	B2
TP15	B3	B3
TP16	B4	B4
TP17	B5	B5
TP18	B6	B6
TP19	B7	B7
TP20	B8	B8

Table 3.4 Bock identification results for test points on Upper floor (F2)

3.3.2.2 Distance estimation results

The position within a block is calculated using lateration technique, which requires estimation of distances to Wi-Fi beacons. The distances are estimated using the propagation models stored on database. At each test point multiple Wi-Fi beacons

are scanned and distances to each of these Wi-Fi beacons is estimated. The average of distance estimation errors for each test point is illustrated in figure 3.23 as a CDF. The 50th percentile is 2.7m and the 90th percentile is 10m.

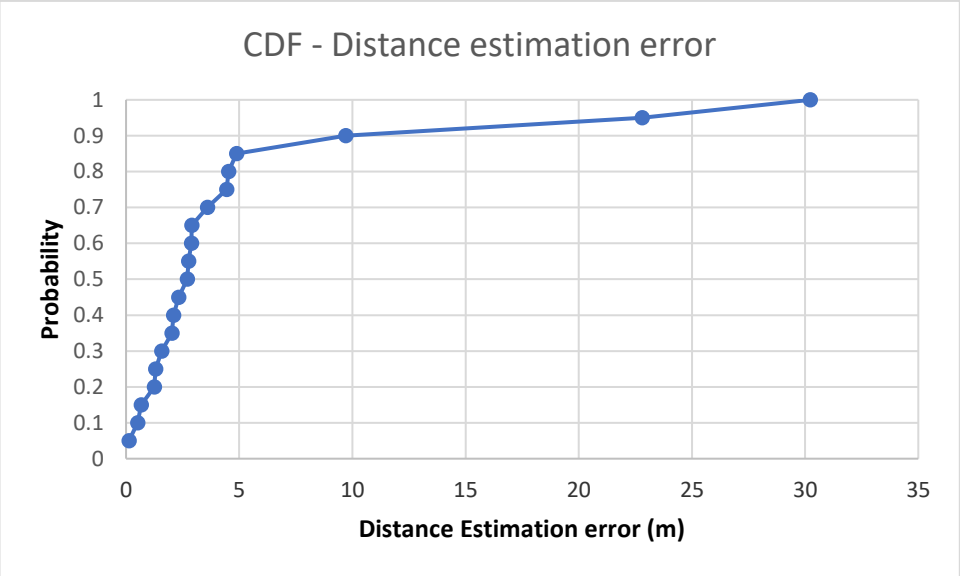


Figure 3.23 Distance estimation error (CDF)

3.3.2.3 Lateration results

The three-dimensional lateration technique used in this initial implementation did not provide any reasonable results. The reasons for the failure of lateration technique are analysed in the next section.

3.4 Analysis

This section analyses results from various components of currently proposed system. The analysis covers propagation modelling, database size, fingerprint and lateration algorithm used in proposed system.

3.4.1 Analysis of propagation modelling and distance estimation:

Section 2.6.2 discussed wall and floor path loss model and ITU models. Wall and floor propagation model [60] use fixed propagation co-efficient (γ) and adds factors for other propagation losses such as the walls and floors. ITU model [62] considers propagation co-efficient as a variable to account for losses between transmitter and receiver on same floor and adds factors for losses due to floors.

For the current proposed propagation modelling, propagation co-efficient is considered to be a variable to account for both free space and environmental losses.

Figure 3.19 shows a plot of RSS and log-distance for Wi-Fi beacon "00:13:5F:F8:F3:F0". A linear regression is fitted for plot, which gives a propagation model for Wi-Fi beacon "00:13:5F:F8:F3:F0" in block 4 on upper floor (F2). The variation of RSS with distance is shown in figure 3.24.

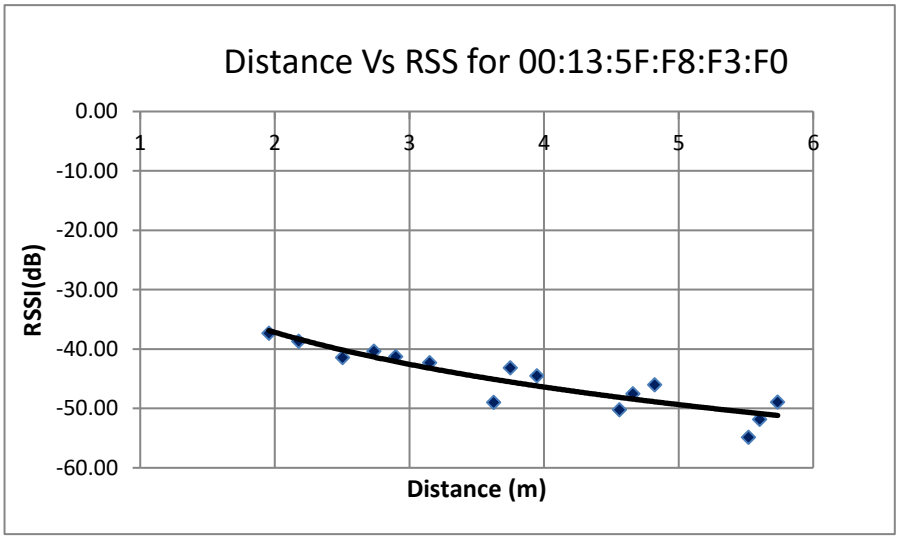


Figure 3.24 Plot of RSS vs Distance for Wi-Fi beacon "00:13:5F:F8:F3:F0"

According to radio propagation theory discussed in chapter 2, RSS is inversely proportional to square of the distance. In figure 3.24 distances from the Wi-Fi beacon increases from 1.9m to 5.7m and corresponding RSS decreases from -37dB to -54dB, which is in accordance with free space propagation theory. In indoor environments due to environmental obstructions, reflection, multipath and fading effects are prominent in some areas than others. So, in the areas where these effects are significant RSS is not always inversely proportional to distance. This is illustrated using Wi-Fi beacon "00:12:44:B3:31:70" and its RSS variation with distance in block 4 on upper floor (F2).

Figure 3.25 shows a plot of RSS and distance variation for Wi-Fi beacon "00:12:44:B3:31:70" in block 4 on upper floor (F2). Distance varies between 7.5m to 10m and RSS between -82dB to -86dB. The plot shows that RSS increases with distance in block 4 on upper floor (F2) for Wi-Fi beacon "00:12:44:B3:31:70".

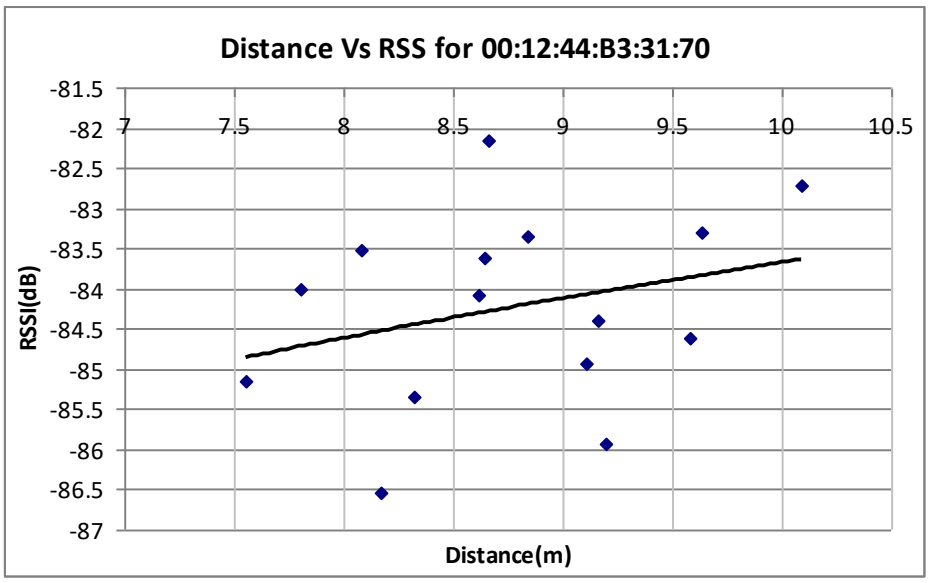


Figure 3.25 Plot of Distance Vs RSS for Wi-Fi beacon 00:12:44:B3:31:70

The increase in RSS with distance in this particular scenario is due environmental effects. Area where the data was collected is close to a stair case. As the distance from the Wi-Fi beacon increases, the open space in the stair well provides fewer obstructions and also due to multipath effect, the RSS measured is higher. This example shows the significant impact a propagation environment has on RSS.

To analyse RSS variation with distance, data collected for the five know Wi-Fi beacons during calibration phase is plotted. *Figure 3.26* shows 10*Log(d) Vs RSS plot for Wi-Fi beacon "00:1B:8F:88:9C:20". The plot uses data collected throughout the test area and also indicates block in which the Wi-Fi beacon was collected.

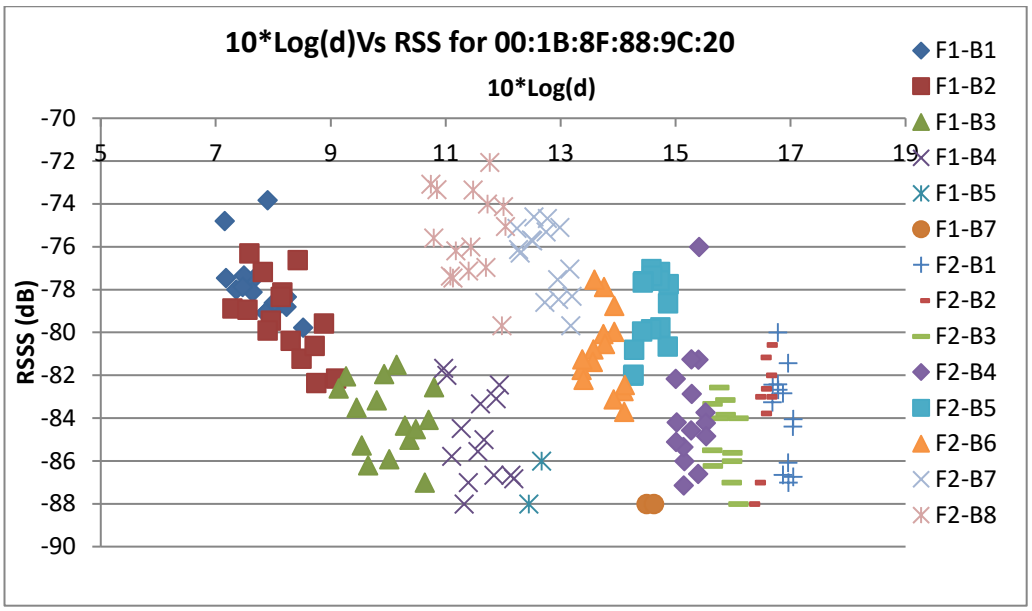


Figure 3.26 Plot of 10*Log(d) Vs RSS for Wi-Fi beacon "00:1B:8F:88:9C:20"

A similar plot in a free space environment with no obstructions would be a straight line. The plot in *figure 3.26* shows that RSS variation with distance and with environmental effects is much more complicated. This is also evident from plots of data for other known Wi-Fi beacons shown in figure 3.27.

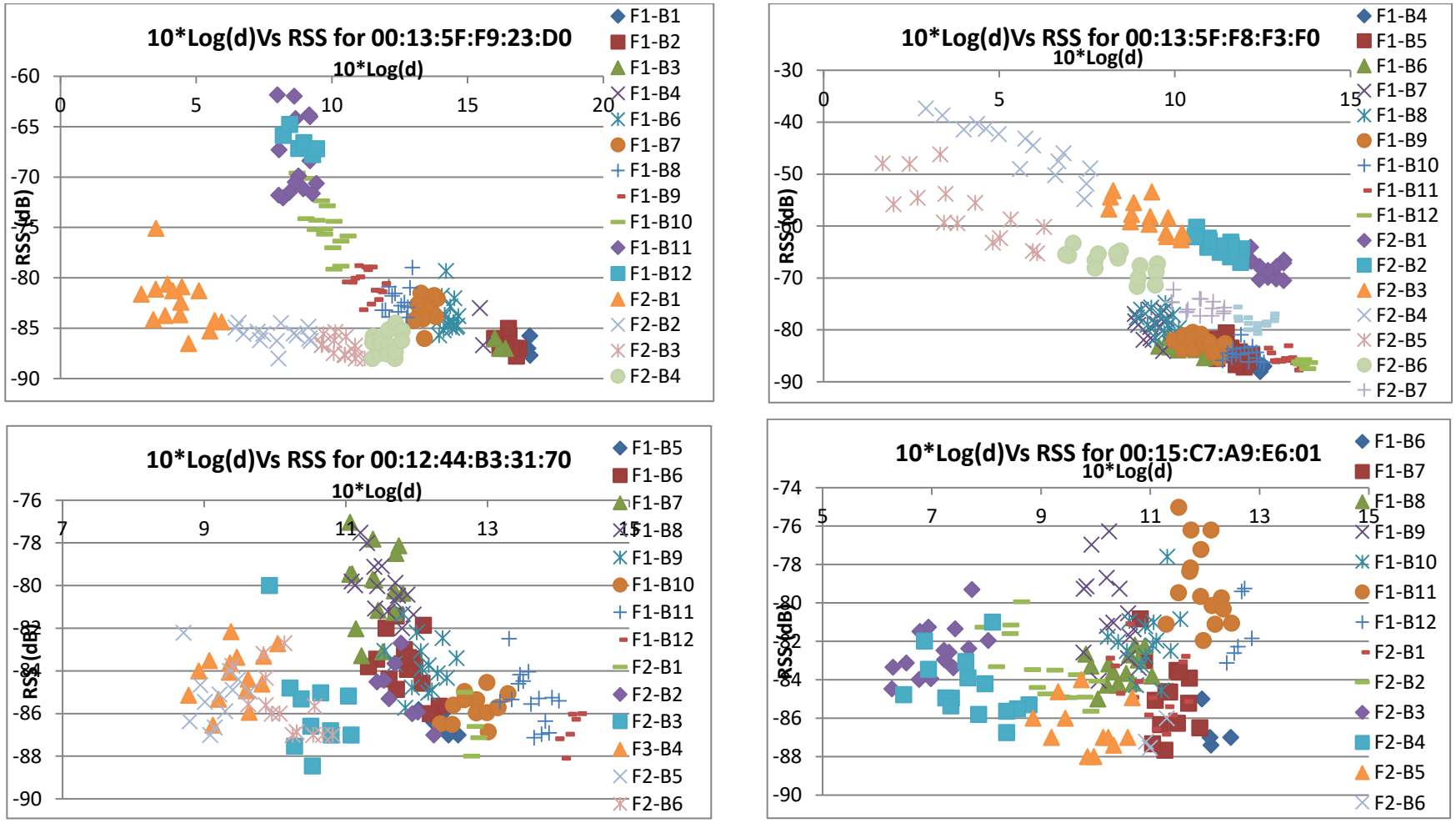


Figure 3.27 Plot of 10*log(d)Vs RSS for known Wi-Fi beacons

From figure 3.27, Consider the plot for “00:13:5F:F8:F3:F0” which is used below to analyse the trends observed and to correlate them to the floor plan. Figure 3.32 shows the log-distance Vs RSS plot with trends highlighted in Blue, Green and Orange. The corresponding blocks on floor plan of the upper floor (F2) are also highlighted in the same colour scheme and the location of beacon is also shown.

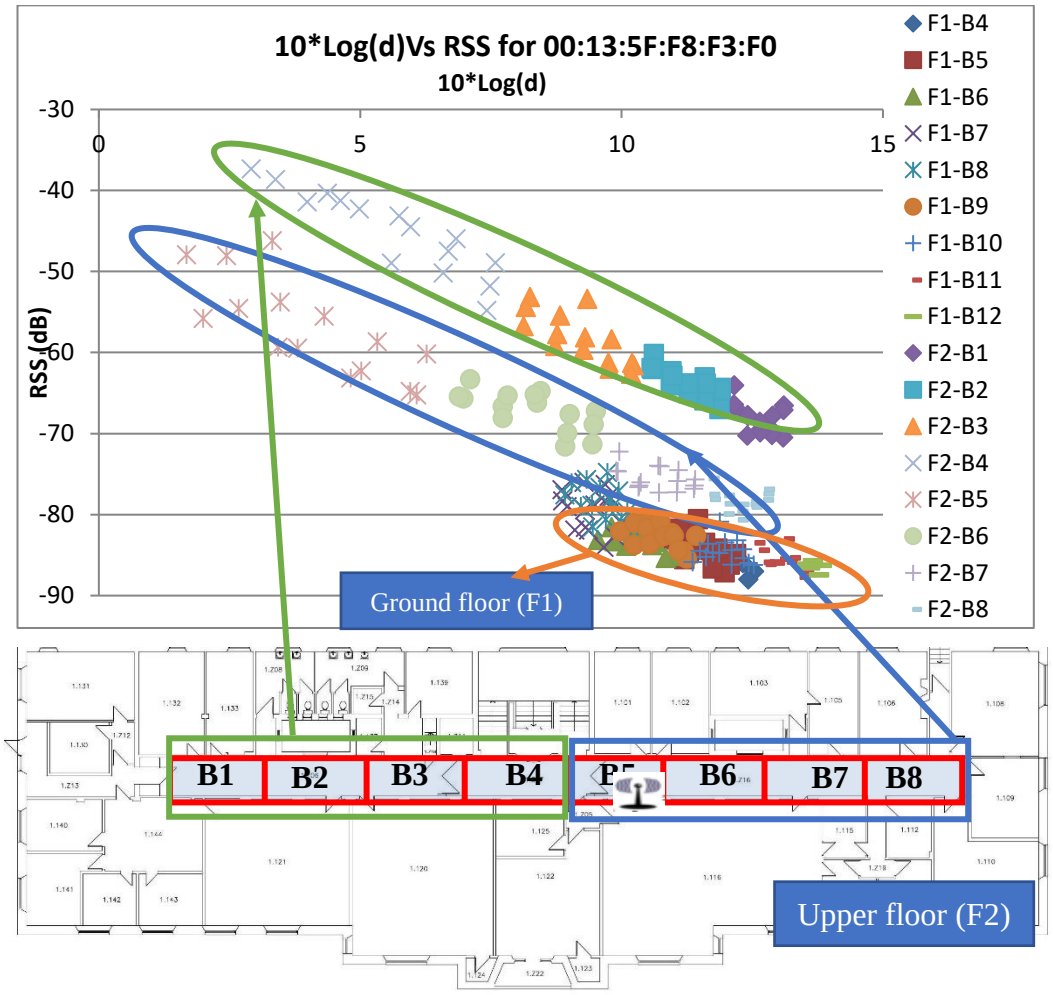


Figure 3.32 Plot of 10*Log(d) Vs RSS for Wi-Fi beacon “00:13:5F:F8:F3:F0” indicating corresponding blocks

Blocks 5 to 8, are to the right of the beacon and blocks 1 to 4 are to the left. The corresponding plot of log-distance vs RSS of the calibration data collected in blocks on left hand side form a distinctive line highlighted in green and the calibration data collected in right hand side blocks, form another line highlighted in blue. Finally, the calibration data collected on ground floor (F1), forms another shorter cluster highlighted in orange. Similar lines can be observed in other beacon’s

log-distance vs RSS plots in figure 3.27. These lines indicate that environmental factors create patterns in RSS variation with distance in an indoor environment.

Another aspect to consider is the accuracy of distance estimation using block models. Looking at the CDF shown in figure 3.23, the 85th percentile of distance estimation error is less than 5m and three test points had average error estimations of approximately 10m, 22m and 30m. The distance estimation errors at these test points is larger than the block size even though the correct block was identified. These large errors in distance estimations are due to the usage of a simple propagation model (linear regression). For instance, consider TP11, which has an average of 30m distance estimation error. One of the beacons detected at TP11 was, "00:13:5F:F9:23:D0" with RSS "-71". Using the propagation model for block 11, "y = -0.3424x - 65.056", the error in distance estimation is 54.4m. A plot of log-distance vs RSS for beacon "00:13:5F:F9:23:D0" at block 4 is shown in figure 3.33.

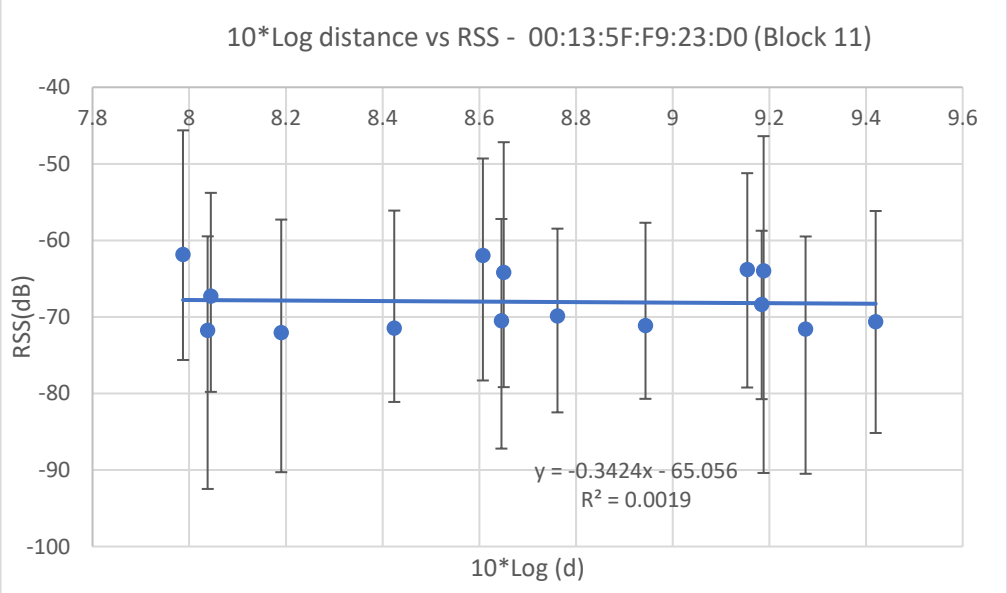


Figure 3.33 Log-distance vs RSS plot for 00:13:5F:F9:23:D0 at Block 11

Figure 3.33 shows the "R^2" which can indicate the goodness of fit, which is 0.19% for the current fit. Such a low R^2 indicates that the variations between log-distance ad RSS are not captured in the line fitted for the data. Scenarios like these can be avoided by discarding or not using the linear regression functions with low goodness of fit indicators such as R^2. The current work has not implemented this goodness of fit criteria and uses all linear regression equations.

Propagation models discussed in previous chapters have a single model that describes relationship between RSS and distance for particular environmental conditions such as an urban environment or a building. If all propagation errors are accounted into propagation co-efficient, a straight line can be considered as a propagation model for whole test area. A straight line fit for Wi-Fi beacon "00:1B:8F:88:9C:20" is shown in figure 3.28 which represents a propagation model for the particular beacon in the current test area.

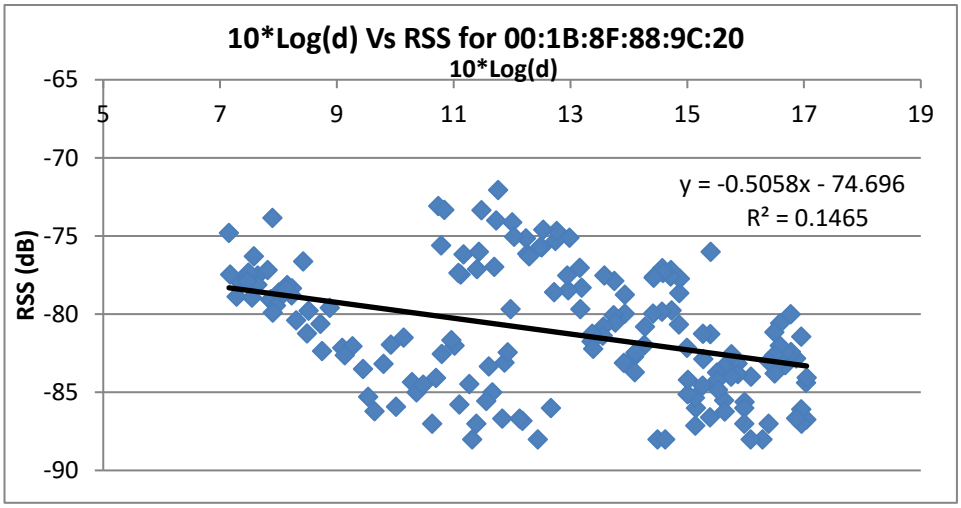


Figure 3.28 Linear fit for plot of 10*Log(d) Vs RSS for Wi-Fi beacon "00:1B:8F:88:9C:20"

The model built using this technique is not a good fit which is evident from figure 3.28. But other propagation models need specific details of obstructions between transmitter and receiver such as number of walls and number of floors, such information is not available to mobile device at an unknown location during positioning phase.

Propagation models are used for distance estimation during positioning phase. These estimated distances determine the accuracy of lateration process. The effect of using a model built for each block (Block model) compared to a model for the whole test area (Building model) can be seen with results in figure 3.34. The test points in which Wi-Fi beacon "00:1B:8F:88:9C:20" was scanned are used for distance estimation using models realised for blocks in the test area. Figure 3.34 shows the errors in distance estimation for Wi-Fi beacon "00:1B:8F:88:9C:20" using a Block model and Building model at test points. The 90th percentile error using

Block model is 4.3m whereas using Building model is 17m. This shows that there is 73.5% improvement to using Block model compared to using Building model.

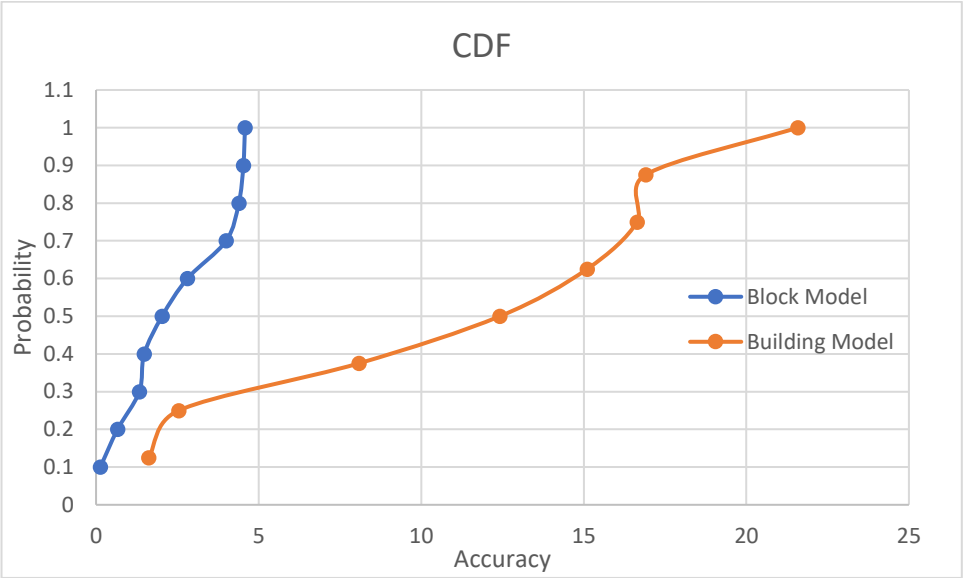


Figure 3.34 Distance estimation errors using Block model and Building model for Wi-Fi beacon "00:1B:8F:88:9C:20"

3.4.2 Analysis of fingerprinting algorithm:

Fingerprinting techniques were discussed earlier in chapter 2.3. Initially, a fingerprint technique implementing nearest neighbour algorithm was implemented for proposed hybrid system. Nearest neighbour algorithm does not perform well in cases where one or more Wi-Fi beacons are missing from the position request scan. A basic implementation of the nearest neighbour algorithm did not provide any accurate block identification. It is statistically impractical to find an exact match [76] of position request to database scans. Another issue with fingerprinting algorithms is that it takes a long time to find a match in test area with dense fingerprints and large number of Wi-Fi beacons.

Data was collected at London shopping mall (Westfield Shepard's bush) using sensewhere's android application. The aim was to analyse number of Wi-Fi beacon matches and time taken to find a match in a large fingerprint data. The implementation of nearest neighbour algorithm was modified to compensate for the missing Wi-Fi beacons. A penalty is added to signal distance calculations based on the RSS of the missing Wi-Fi beacons. If a Wi-Fi beacon with strong RSS either in position request scan or database scan is missing a higher penalty is added and a

lower penalty for weaker RSS. The 161 points at which fingerprint data was collected are shown in *figure 3.29*. After collecting fingerprint data, a test scan which can be used as a positioning request was collected close to fingerprint data point 25.

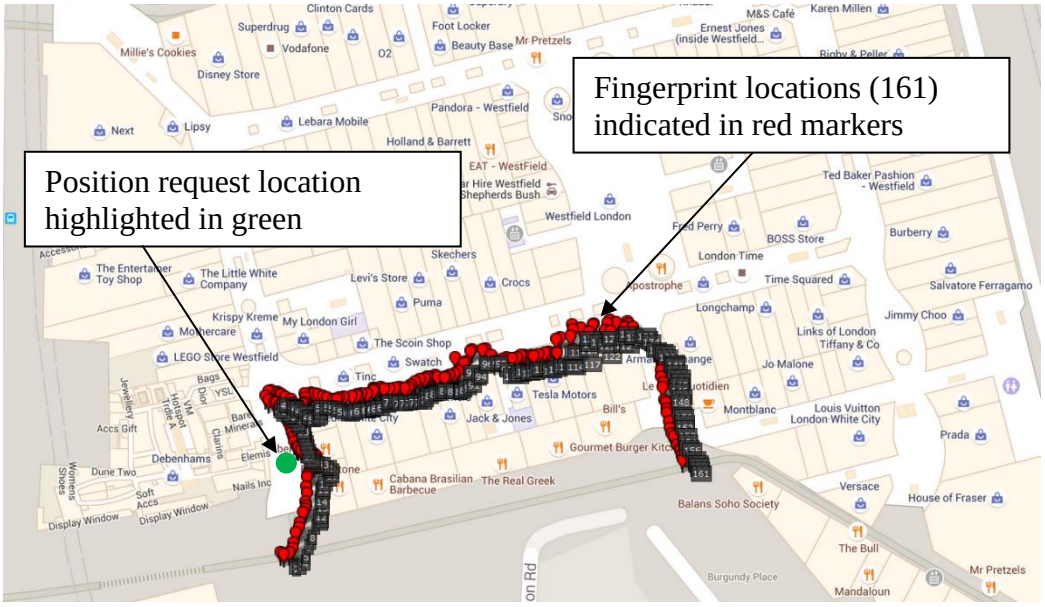


Figure 3.29 Fingerprint locations and position request location in test area

The current implementation of nearest neighbour algorithm logs number of matching Wi-Fi beacons at each fingerprint location between position request and fingerprint scan. Percentage of number of matching beacons at each fingerprint location is shown in figure 3.30.

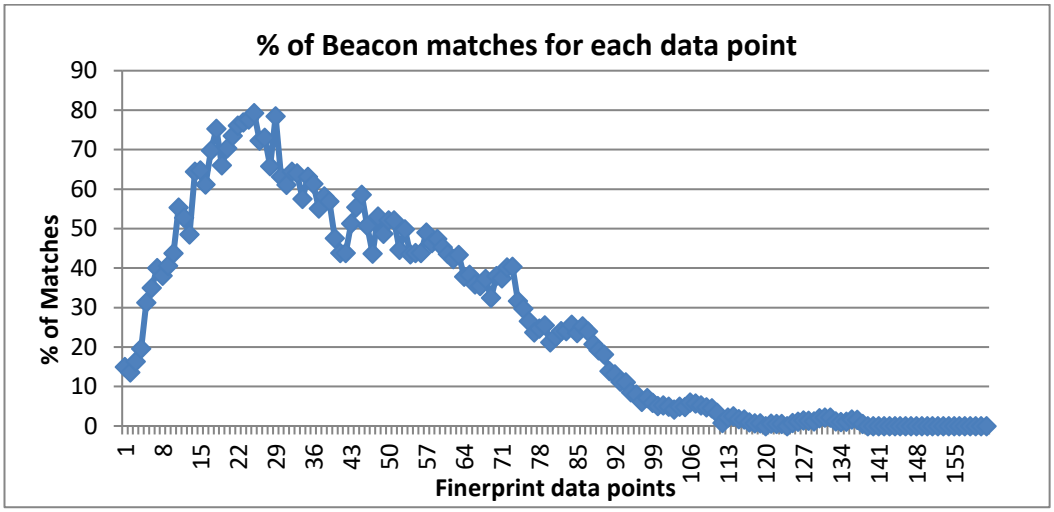


Figure 3.30 Plot of percentage of beacon matches at each fingerprint location

Figure 3.30 indicates the maximum match percentage is 79.2, at fingerprint location number 25. At fingerprint location 25, there were 222 unique Wi-Fi beacons scanned in fingerprint scan and location request scan combined. Of these 222 Wi-Fi beacons, 176 were scanned in both fingerprint and location request scans. This highlights the issue of missing Wi-Fi beacons which would influence the Euclidean distance being calculated. The current implementation of Euclidean distance adds a penalty for missing beacons based on RSS. *Figure 3.31* shows a plot of Euclidean distances calculated at 161 fingerprint locations.

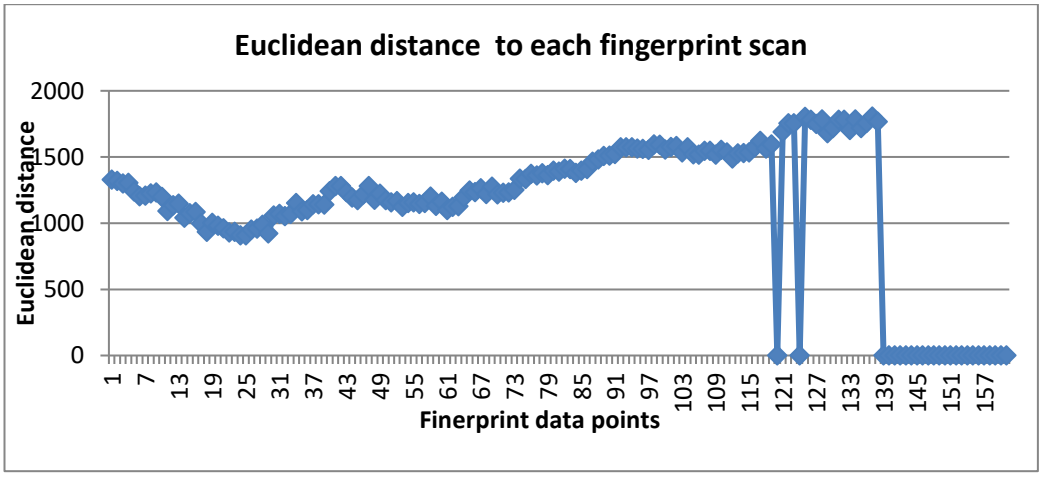


Figure 3.31 Plot of Euclidean distance at each fingerprint location

Nearest neighbour algorithm implementation calculates Euclidean distance described by equation 2.3. The zero Euclidean distances seen in figure 3.30 are for the cases where there were no beacon matches in location request and fingerprint scans. Ignoring these zero Euclidean distance locations, it can be noticed that the smallest distance is at fingerprint location 25. The plot in figure 3.31 also shows increasing Euclidean distances increase moving away from location 25. This shows that addition of penalties for missing beacons has been effective.

The time taken for current implementation to calculate Euclidean distances to 161 fingerprints is about 10mins. This highlights the second issue with fingerprinting algorithms which was computational intensity. Due to the high fingerprint density and also about 200 Wi-Fi beacons in each scan, time taken to calculate the distances are very high. This poses a major issue when providing a location in real time. The current implementation can be optimised by using efficient sorting algorithms but a

long delay in returning estimated position would mean that user could have moved significant distance by the time a response is received from positioning system.

The current test area at Sanderson building has 294 fingerprint locations and each scan has no more than 15 Wi-Fi beacons. So, a fingerprint algorithm would return a result in reasonable time, but a fingerprint solution is not scalable. Using the proposed hybrid system, number of Fingerprint scans were reduced down to 20 (one per each block), thereby keeping the computation time in an acceptable range.

3.4.3 Analysis of lateration algorithm

Lateration algorithm implemented for current system was to linearize the system of sphere equations and to solve them as described in chapter 2. The current implementation failed to provide any reasonable results due to following reasons:

1) The number of beacons required for a three-dimensional lateration is four. Since the total number of Wi-Fi beacons with known locations is five, some areas do not have enough coverage of Wi-Fi beacons to apply lateration algorithm.
2) The distances between the Wi-Fi beacons and mobile device are estimated and have errors. Hence the current implementation of linearizing the system of equations and solving them failed to find intersection points.

3.5 Conclusion

This chapter describes the architecture of proposed hybrid positioning system. The section 3.2 of this chapter describes calibration phase and positioning phase of proposed hybrid system. Calibration phase involves processing collected data to realise propagation models for each Wi-Fi beacon in each block. The processing steps and methods involved were detailed. Block identification and distance estimation processes during positioning phase were described in detail.

Section 3.3 describes the implementation of proposed hybrid system at a university building. Calibration points and data collection process was explained. Realisation of propagation models for each block using linear regression method for multiple Wi-Fi beacons was detailed. A comparison of data size that needs to be stored between proposed hybrid and fingerprinting systems shows reduction by a factor of eight. In later part of this section, steps involved in estimating position of a mobile device are described. 20 test points were selected in the test area and the

position estimations were made. The first step of block identification results shows an accuracy of about 80%. Distances to the beacons were calculated using propagation models corresponding to the identified block. The 90th percentile distance estimation error for 20 test points was about 10m.

Section 3.4 analyses results obtained in the previous section. A comparison of using a single propagation model and block level propagation models was done as part of analysis. The comparison shows about 73% improvement in distance estimations using block level propagation model compared to a building level model. The log-distance vs RSS plot for beacon “00:13:5F:F8:F3:F0” shows three distinctive lines which two correspond to left and right side of the beacon and the third for data from lower floor (F1).

Nearest neighbour fingerprinting algorithm was implemented which adds penalties for missing Wi-Fi beacons was implemented. The performance was analysed at a large shopping mall with high Wi-Fi beacons density. Nearest neighbour algorithm is computationally intensive and takes considerable time to find a match which is not ideal for a positioning system. The reasons for failure of lateration algorithm are listed towards the end.

Chapter 4

Automatic Block Creation

4.1 Introduction

The previous chapter introduces the idea of a hybrid indoor positioning system. The calibration data is collected during offline phase as in other fingerprinting systems. However, the area of interest is then divided into blocks and then the calibration data is used for building a propagation model for each block. The size of data that needs to be stored using the approach presented in the previous chapter has been cut down by a factor of eight. Also, the overall error in distance estimation using propagation model specific to a block is below 5m, 85% of the time. As discussed in section 3.4.1, propagation models specific to each block describe the signal propagation for indoor environments better than models built for whole building. This chapter aims to identify an optimal block size and evaluate if the optimal block size can be identified automatically from calibration data.

4.2 Block Size and Distance estimation

In most cases, block-based propagation models manage to capture the signal variation with distance, however in some scenarios the signal variation in small over the distances covered by each block (5m × 2.5m). For example, the log-distance vs RSS plot shown in figure 3.33 for beacon "00:13:5F:F9:23:D0", RSS fluctuates up to 10dB, but the change in RSS is not due to variation in distance (since R^2 is .19%). Hence, when a linear regression is applied, the resulting line

does not encapsulate the variation of RSS and distance. One possible reason for only minor variation in the RSS could be the current block size, which is 5m × 2.5m, being too small. Distance estimation using linear regression model in such scenarios results in large errors.

Another advantage of block size variation can be noticed in figure 3.32, where three distinctive lines can be seen from the plot of log-distance vs RSS. It is possible to change the block size for beacon “00:13:5F:F8:F3:F0”, such that the entire area of interest covered by the beacon is divided into only three blocks. First would be a single block covering the left-hand side, second, the right-hand side and third, covering entire ground floor (F1) of the area of interest. This will reduce the amount of data that needs to be stored and also there by the computational intensity.

Based on the two scenarios discussed above, block size can be changed to reduce the amount of data that needs to be stored. The following section analyses the effect of varying block size on distance estimation.

4.3 Effect of Block size on Distance Estimation

Size of the block influences the accuracy of distance estimation and also the amount of data that needs to be stored. To analyse this impact on distance estimation, the calibration data collected for each beacon is divided into blocks of 10m,15m and 20m and compared with the distance estimation accuracy with 5m block size. The distribution of different block sizes is shown in figure 4.1.

Figure 4.1 shows the floor plan of ground floor (F1) and upper floor (F2) along with corresponding blocks in different sizes. The block sizes are made larger in increments of 5m. The number of blocks on ground floor (F1) reduces from 12 with 5m, to 6 with 10m, to 4 with 15m and to 3 with 20m. On upper floor (F2), the number of blocks change from 8 with 5m, 4 with 10m, 3 with 15m and 2 with 20m. However, one of the three blocks (B3) with 15m block size, which was supposed to be 15m is only 10m, due to the size of the area of interest.

After the size of the blocks is identified, data from each block is used to generate a propagation model for each beacon. Then, RSS from test point scans are used to estimate the distances for each beacon, to evaluate the accuracy of distance estimation.

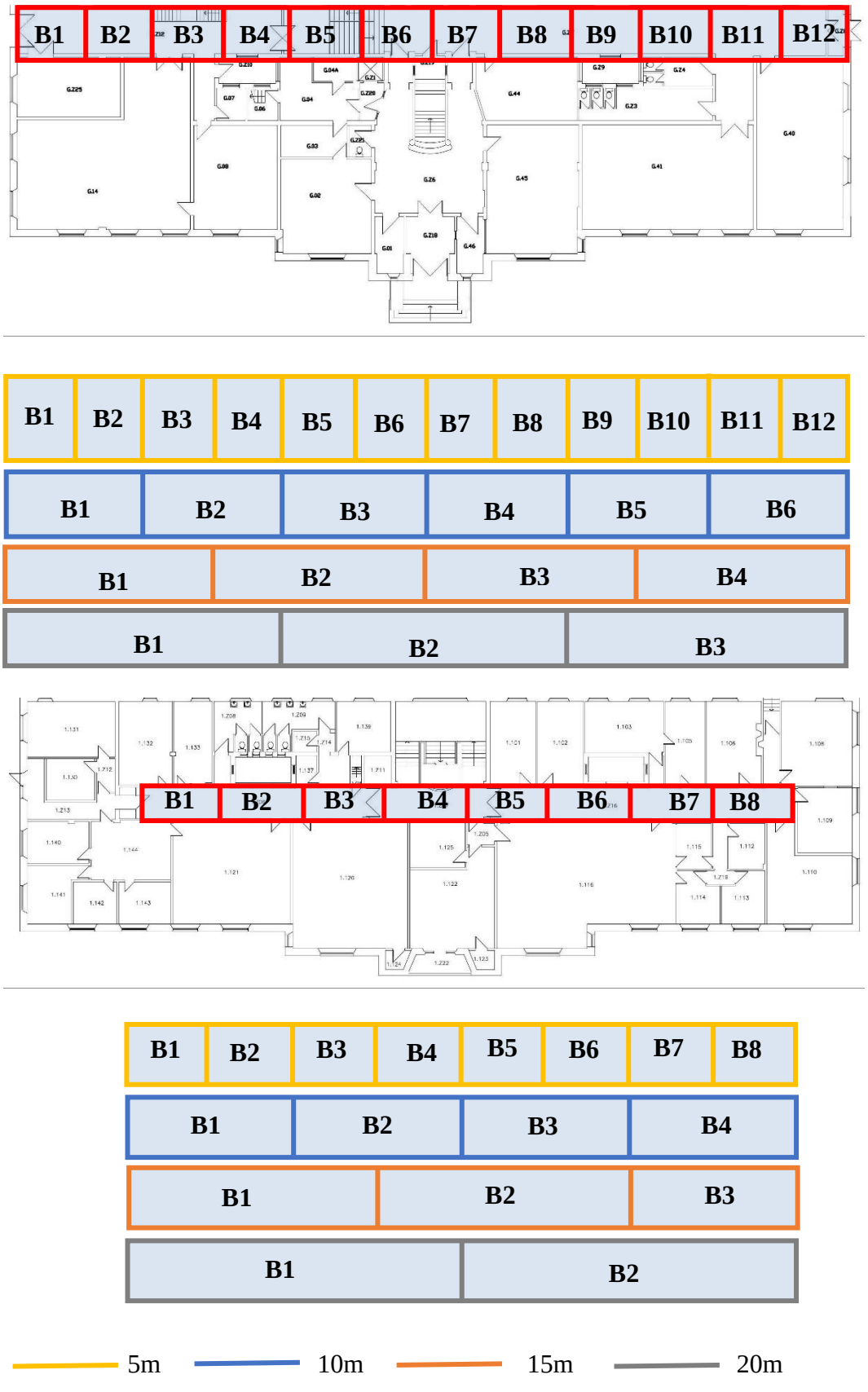


Figure 4.1 Floor plans and various block sizes

Signal modelling was done for all five beacons in the test area identified in table 3.1, with different block sizes as shown in figure 4.1. Distance estimation errors for each beacon at all test points is calculated and CDF of the distance estimation errors is shown in the following figure 4.2.

Beacon “00:13:5F:F8:F3:F0”, propagation was discussed earlier where the log-distance vs RSS plot created distinct patterns for left and right of the beacon. This formed a basis for the current block size analysis. So, CDF of the distance estimation error for “00:13:5F:F8:F3:F0” is shown in figure 4.3 for various block sizes. The best performance is with a 5m block size, where the 90th percentile is approximately 3m. The 90th percentile with 10m block size and 15m block size is approximately 5m. Finally, for a 20m block size, the 90th percentile distance estimation error is approximately 7m.

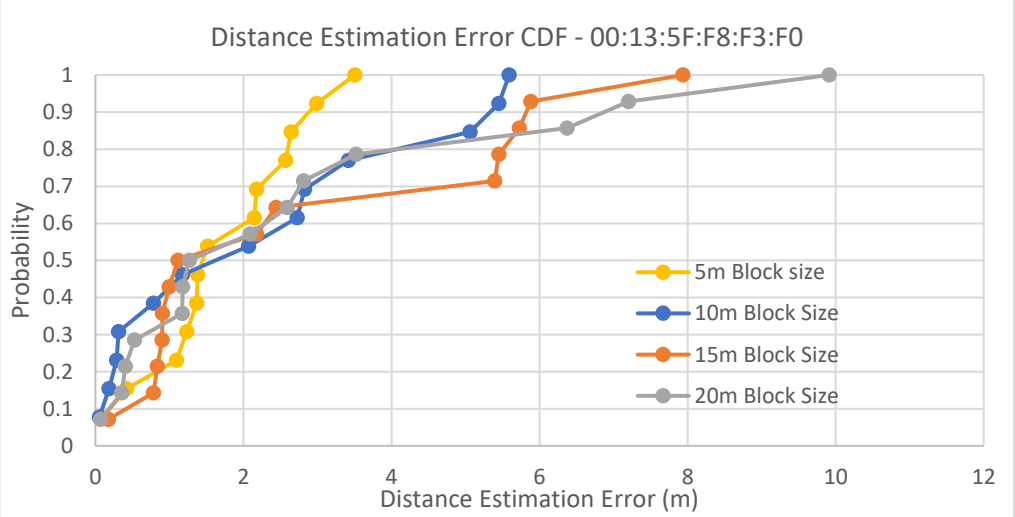


Figure 4.3 CDF of distance estimation error for “00:13:5F:F8:F3:F0” with different block sizes

However, in figure 4.3, below 50th percentile, 10m block yields the best performance and above 50th percentile, 5m has the best performance with 90th percentile less than 3m distance estimation errors. It is not intuitive to compare 50th percentile and 90th percentile, but for applications where accuracy is not the only requirement the comparison helps. In systems where the best accuracy is returned only 50% of the time and is yet acceptable, 10m block size would be a better option to consider as the amount of data to be stored would be halved and also reduces the computational intensity.

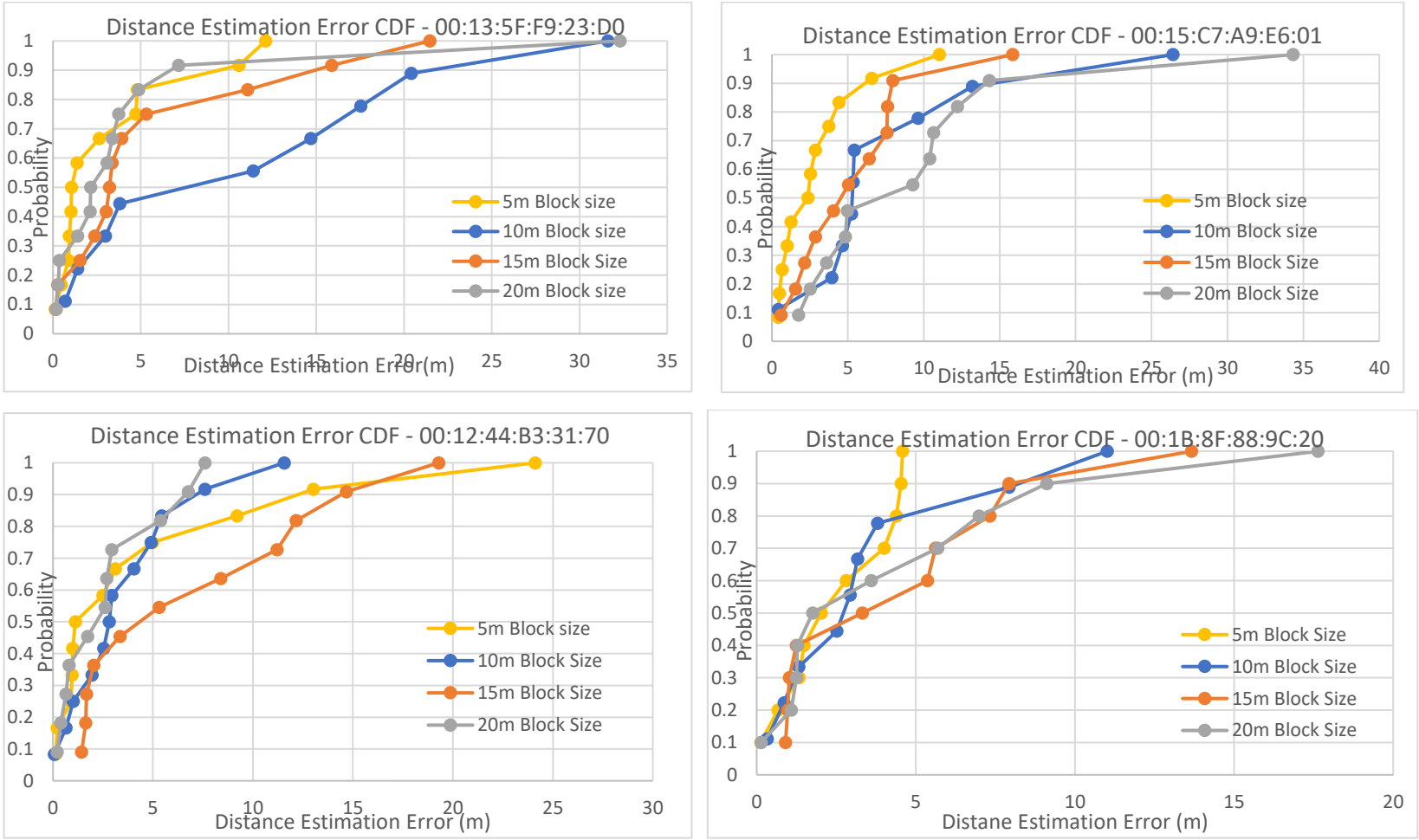


Figure 4.2 CDF of Distance estimation error with different block sizes

A Comparison of best distance estimation (90th percentile) for each beacon in the test area and corresponding block size is shown in table 4.1. A single block size does not provide the best distance estimation. It is also interesting to find that the smallest and largest block sizes give the best estimation. Because, in case of "00:13:5F:F8:F3:F0", a bigger block size could have captured the propagation into three blocks as highlighted in figure 3.2 which could provide better fit for linear propagation models.

Beacon (MAC id)	Best distance estimation block size (m)	90th percentile (m)
00:13:5F:F8:F3:F0	5m	3m
00:13:5F:F9:23:D0	20m	7m
00:15:C7:A9:E6:01	5m	6m
00:12:44:B3:31:70	20m	6m
00:1B:8F:88:9C:20	5m	4m

Table 4.1 Block size and distance estimation accuracy

4.4 Analysis of Block size and Distance Estimation

As noticed in the previous section a single block size does not provide the best distance estimation. In majority of the cases 5m block size seems to give a better distance estimation. However, varying the block size does not always improve distance estimation. The main reason, a fixed size grid like block size does not provide the best distance estimation, is because this approach does not consider the propagation environment.

Consider the scenario where a regular sized division of area into blocks leads to "Block A" (as in figure 4.4), which includes a wall and a beacon is located on one side of the wall. In this hypothetical case, with calibration points on either side of the wall, a linear propagation model for "Block A" would not be able describe the variation of RSS with distance. However, if "Block A" is divided into further blocks such that the wall is avoided, in this case "Block B" and "Block C", a linear propagation model would be able to describe the RSS variation with distance better. In "Block B" since the beacon would have line of sight (LOS) to the mobile device, a linear propagation model would fit. Also, in Block C, the signal would be

attenuated because of the wall and a sperate propagation model for “Block C” would be a better fit compared to a single propagation model for “Block A”.

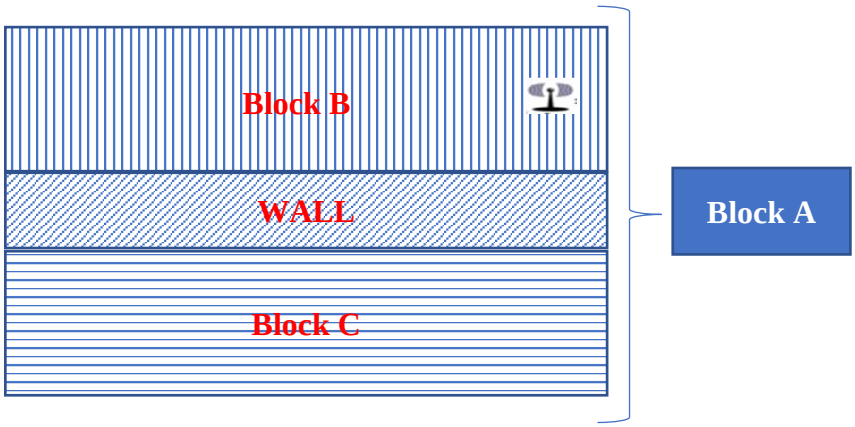


Figure 4.4 Environmental factors and Block size

Another scenario that can describe and help understand the drawback of regular block sizes is that of beacon “00:13:5F:F8:F3:F0” discussed in section 3.4.1. The Log-distance vs RSS plot for beacon “00:13:5F:F8:F3:F0” with 5m blocks is shown here again in figure 4.5.

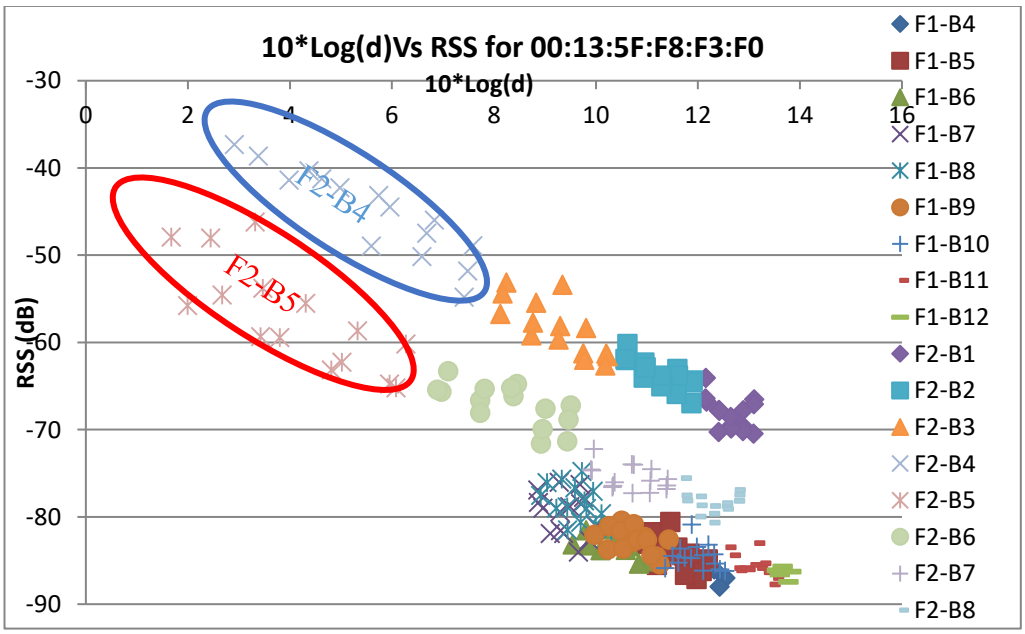


Figure 4.5 Log – Distance vs RSS for “00:13:5F:F8:F3:F0” with 5m blocks

Figure 4.5 highlights two 5m blocks, “F2-B4” in blue and “F2-B5” in red. These two blocks are adjacent to each other as highlighted in figure 3.32. When picking a 10m block, “F2-B4” and “F2-B5” could form a single block as they are

adjacent. If that happens, it is clear from the plot in figure 4.4 that a linear propagation model fitted for the 10m block would be worse compared to individual 5m blocks.

4.5 Alternative block size determination

As discussed in the earlier section, regular sized blocks cannot provide an accurate propagation model all the time, as the environmental factors do not form part of the block size determination criteria. The other alternative for block size determination which takes environmental factors into account is use building floor plan. One of the solutions is to consider each open area (such as rooms) without obstructions (such as walls) as a block. As discussed in previous section in figure 4.4, a block without environmental structures (such as walls) is likely to have a variation of RSS with distance which can be modelled with a simpler linear model compared to other modelling techniques. To determine blocks based on the environmental factors, a floor plan of the test area is required. However, even the knowledge of the layout alone would not help as the location of the beacons within the layout, would create a complicated environment. In some cases, one side of the room would have only one wall between the mobile device and the beacon, while on other side of same room multiple walls could be in the path between mobile device and the beacon. An illustration of such a scenario is shown in figure 4.6.

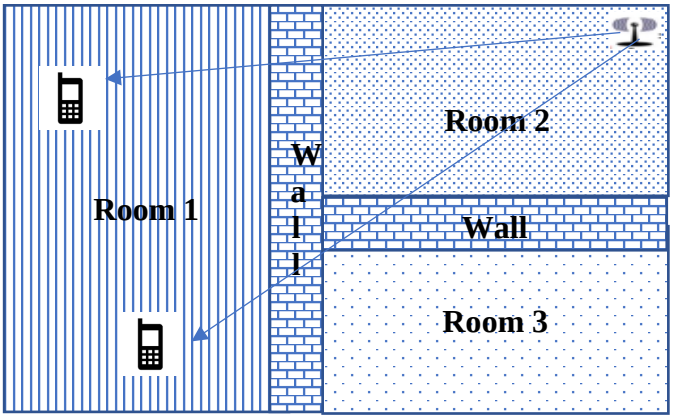


Figure 4.6 Illustration of complex propagation environment

Based on topological block determination, the test area in figure 4.5 can be divided into three blocks, one for each room. In “Room 1”, when the device is in upper part (north) of the room, there is only one wall in between the mobile device

and the beacon. However, in lower half (south) of the room, there would be two walls in between mobile device and the beacon. Hence, even topological division of the test area into blocks would not separate the calibration data, which can be used to model a linear propagation model.

Another solution to separating the calibration data into blocks is to find patterns in calibration data. This can be achieved by using machine learning, clustering algorithms. A brief overview of clustering algorithms is provided and some basic algorithms are applied to beacon "00:13:5F:F8:F3:F0" to evaluate automatic identification of blocks. Clustering can be broadly divided in to two categories, Hard Clustering where each data point either belongs to a cluster or not and Soft Clustering where each data point could belong to multiple clusters (with a confidence associated with each cluster) [174]. A classification of some of the Clustering algorithms is presented in [175] depending on the techniques used in the algorithms as shown in figure 4.7. Partitioning based algorithms, divide the data into clusters and each data point must belong to one cluster. Hierarchical algorithms generate a dendrogram where the data points are divided into clusters and sub-clusters. Density based algorithms cluster the data points depending on the density, connectivity and boundary. Grid based algorithms go through the data points and compute statistical values which are used to cluster the data into grids. Model based systems divide the data points into clusters based on a (predefined) mathematical model. It assumes that the data collected follows underlying probability distributions.

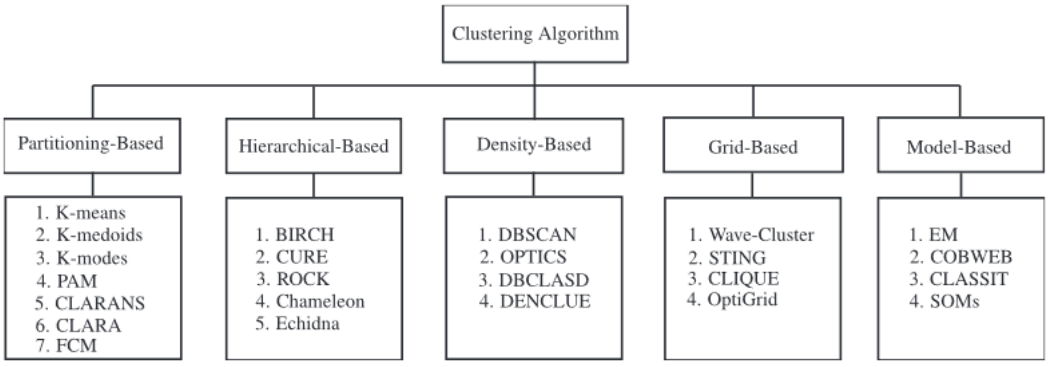


Figure 4.7 Classification of clustering algorithms

The current work does not attempt to study all available clustering algorithms and only attempts to evaluate using of clustering algorithms to log-distance and RSS data to identify optimum block size. The evaluation considers two

clustering algorithms implemented in “Scikit-learn” ([176], [177] and [178]) python library. An example implementation to compare various clustering is shown at [179], displayed here in figure 4.8.

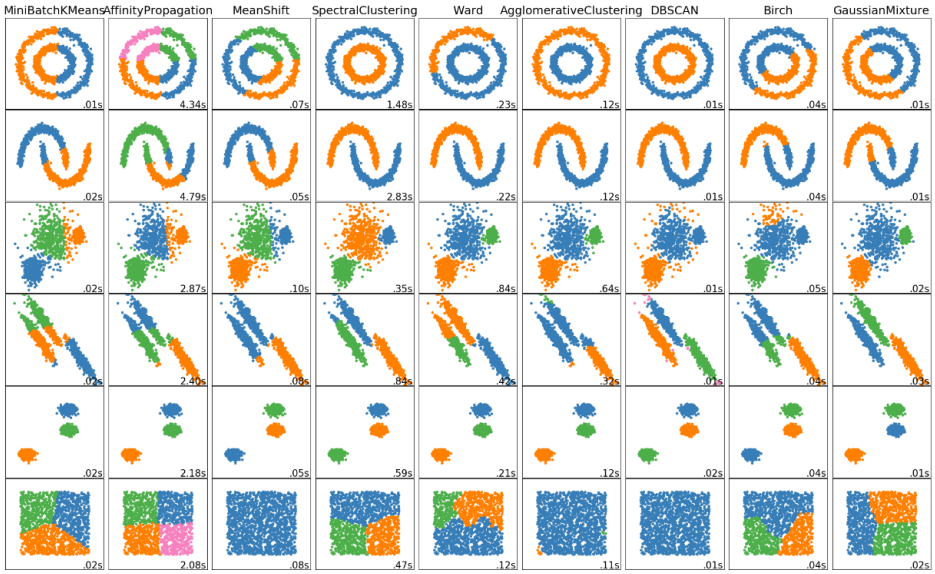


Figure 4.8 Cluster algorithms comparison with Scikit-learn library

For evaluation of using clustering algorithms to identify block sizes, K-means clustering and DBSCAN (Density Based Spatial Clustering of Applications with Noise) algorithms were applied to the data collected for beacon “00:13:5F:F8:F3:F0”. These two algorithms were easier to implement compared to other algorithms, where basic parameters had to be set and the function implemented in the library returns the clusters. So, in the current work clusters generated using K-means and DBSCAN algorithms are evaluated.

K-means algorithm requires the number of clusters as an input. Figure 4.9 shows the results of K-Means clustering applied to data collected for beacon “00:13:5F:F8:F3:F0” with 2, 3, 4 and 5 clusters. With two clusters, most of the data from floor 1 and some of the data from floor 2 (Block 8, refer to figure 4.5) is grouped into one cluster and the rest of data from floor 2 (both left and right of the beacon) is clustered into second. With three clusters, data from floor 1 and some of the data from floor 2 (Block 8, refer to figure 4.5) is still clustered into a single cluster. However, most of data from F2-B4 block and some from F2-B5 is clustered in to a second cluster. With four clusters, data from floor 1 is separated into an individual cluster, however, other clusters have data from overlapping blocks. Similar clustering pattern can be noticed with five clusters well.

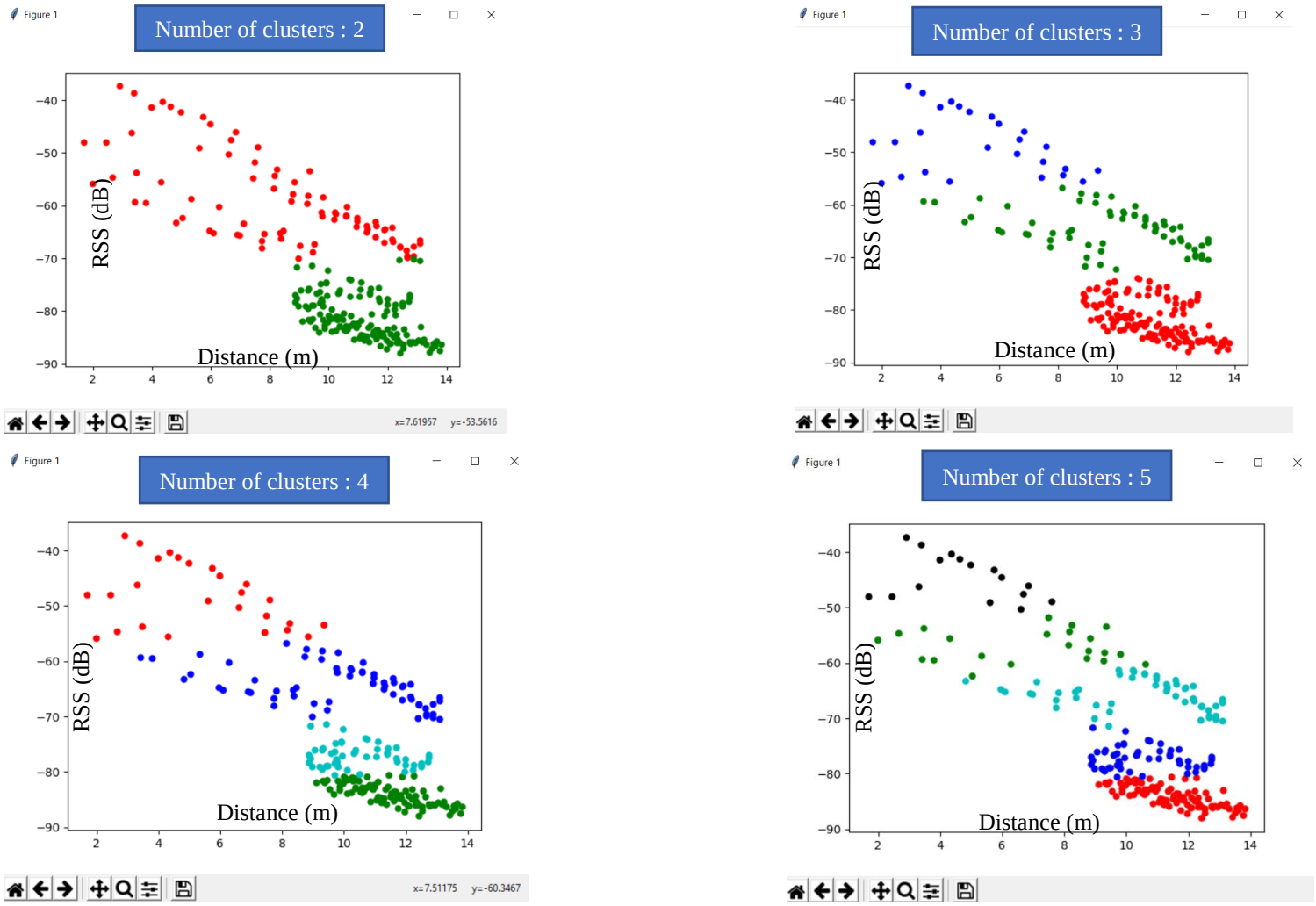


Figure 4.9 K-mean clustering applied to beacon “00:13:5F:F8:F3:F0”

If clusters generated by K-means algorithm are to be used as individual blocks, in most of the cases fitting a linear propagation model would fit. However, in the cases where part of data from left of the beacon and right of the beacon are clustered into one, a linear propagation model would not be an ideal fit. This scenario can be noticed in all cases show in figure 4.9.

DBSCAN, is another clustering algorithm used to evaluate automatic block generation. There are various parameters that can be set and altered to modify the clustering. For simplicity, the current work only focuses on two parameters, first maximum distance between two data points to be considered as in same neighbourhood (represented as “eps”) and second, number of data points in a neighbourhood for a point to be considered as a core point (represented as “min_samples”). A plot of various combinations of “eps” and “min_samples” is shown in figure 4.10. It can be noticed that changing “eps” value from 0.2 up to 0.4 has changed the clusters from three to four. Figure 4.10.1 shows three clusters based on calibration data. These clusters can be compared to figure 4.5 to analyse, data from which blocks contributed to different clusters. Changing “eps” from 0.2 to 0.3(figure 4.10.2) has decreased the cluster size (purple cluster) covering data from left and right side of the beacon. However, it has created a smaller cluster for data from F2-B4. Changing the “eps” value further to 0.4 (figure 4.10.3) has almost eliminated overlap of data between left and right of the beacon. Parameters with “eps” set to 0.4 and “m_samples” set to 3, the overlap has been completely eliminated between left and right side of the beacon. The overlap of blocks from floor 1 and floor 2 shown in red cluster remains in all the combinations, which is due to the proximity and density of data within the blocks in red cluster.

Figure 4.10.1
“eps” = 0.2, m_Samples =4

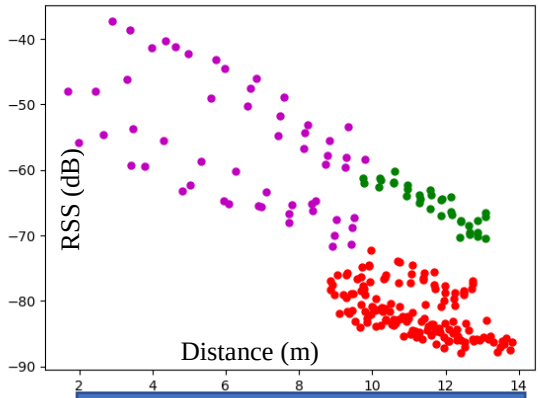


Figure 4.10.2
“eps” = 0.3, m_Samples =4

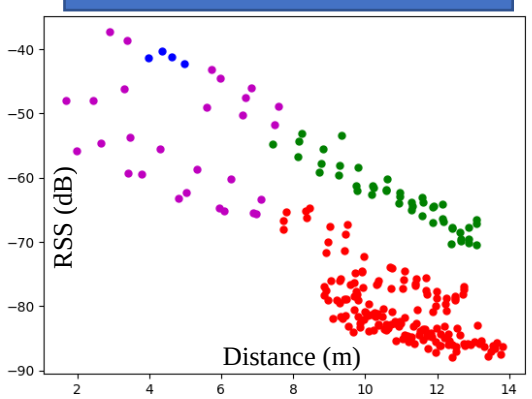


Figure 4.10.3
“eps” = 0.4, m_Samples =4

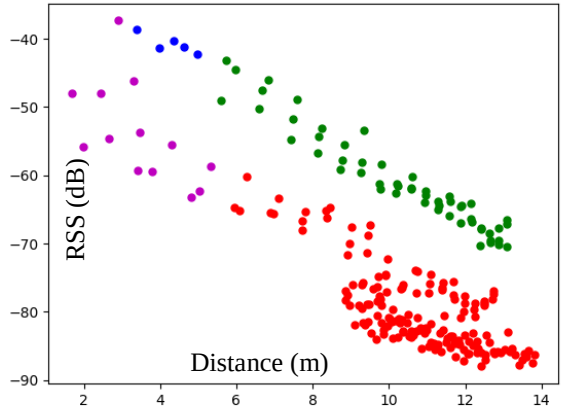


Figure 4.10.4
“eps” = 0.4, m_Samples =3

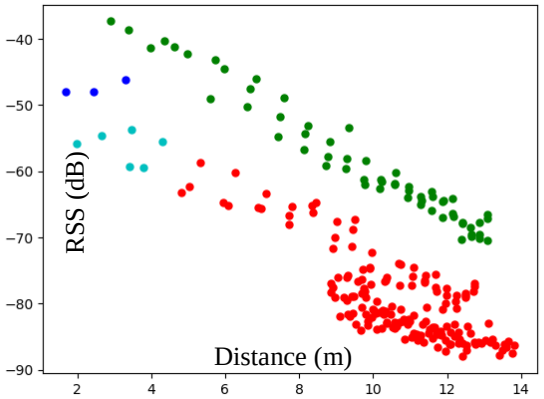


Figure 4.10 DBSCAN clustering applied to beacon “00:13:5F:F8:F3:F0”

Results from DBSCAN clustering indicate that generating the block sizes automatically which can lead to fitting linear propagation models is possible. There are other clustering algorithms and tuning the parameters in those algorithms could eventually generate blocks with different sizes influenced by the environmental factors.

4.6 Conclusion

This chapter analysed the influence of block size on propagation model generation which in turn impacts the distance estimation accuracy. Distance estimation accuracy using regular block sizes varying from 5m to 20m were compared in section 4.3. In most cases (3 out of 5) the smallest block size (5m) provides the best 90^{th} percentile results. However, regular block sizes don't take environmental factors into account when creating the block. For example, a lot of times these blocks could end up spreading on either side of a wall or similar environmental structure (as discussed in section 4.4) so a topological block size which takes environmental structures into account was considered. These topological blocks are evaluated further in the next chapter. But, scenarios where even topological blocks would fail were identified in section 4.5. As an alternative, clustering algorithms were evaluated to automatically identify variable sized blocks. The clustering algorithms implemented (K-Means and DBSCAN), indicate that by tuning the parameters of the algorithms, it could be possible to generate blocks automatically.

Chapter 5

System improvements

5.1 Introduction

The initial implementation of proposed positioning system had some issues discussed in chapter 3. Improvements to following components of the system are proposed

1) Database and Fingerprint algorithm
2) Lateration algorithm

In previous chapters the first implementation of proposed system was discussed and results were analysed towards the end of chapter 3. As a result of the analysis it was clear that current implementation of lateration algorithm was not a good fit for proposed system where estimated distances are subject to errors. An alternate lateration algorithm was non-linear least squares which would account for the errors in distance estimations.

One of the main objectives was for the system to be scalable. And for a system to be scalable, the system should work with geographical coordinate system rather than local coordinate system. But latitude-longitude co-ordinate system grid does not vary evenly on surface of the earth. One-degree difference in latitude and longitude at equator is 111Km whereas moving towards poles a difference of one degree in longitude goes to zero [77]. The usual solution is to transform geographical coordinates to Earth Centred Earth Fixed (ECEF) Cartesian coordinates for ease. But transformed spherical coordinates still have the limitations of being a three-

dimensional system. Hence, a transformation algorithm to convert geographical coordinates to two dimensional local coordinates is proposed in this chapter.

As part of the positioning system, Bluetooth beacons were designed which can transmit location of Wi-Fi beacons and can be read automatically by a compatible device. These beacons can also be installed in areas where Wi-Fi coverage is inadequate. The development of Bluetooth beacons is discussed in later chapter (as part of appendix).

5.2 Database and Fingerprint algorithm

The current system processes collected calibration data for each block before storing to a database. The data stored for each block has two components as shown in *figure 3.5*. First component is the calibration scan collected at the centre of the block and second, propagation models for each Wi-Fi beacon with in the block. In order to reduce database further reduce data to be stored, algorithm is modified to store only propagation models for each block and minimum and maximum values of RSS values used to generate the model, eliminating storage of any calibration scans. Figure 5.1 shows data stored to a database without calibration scans at centre of the block.

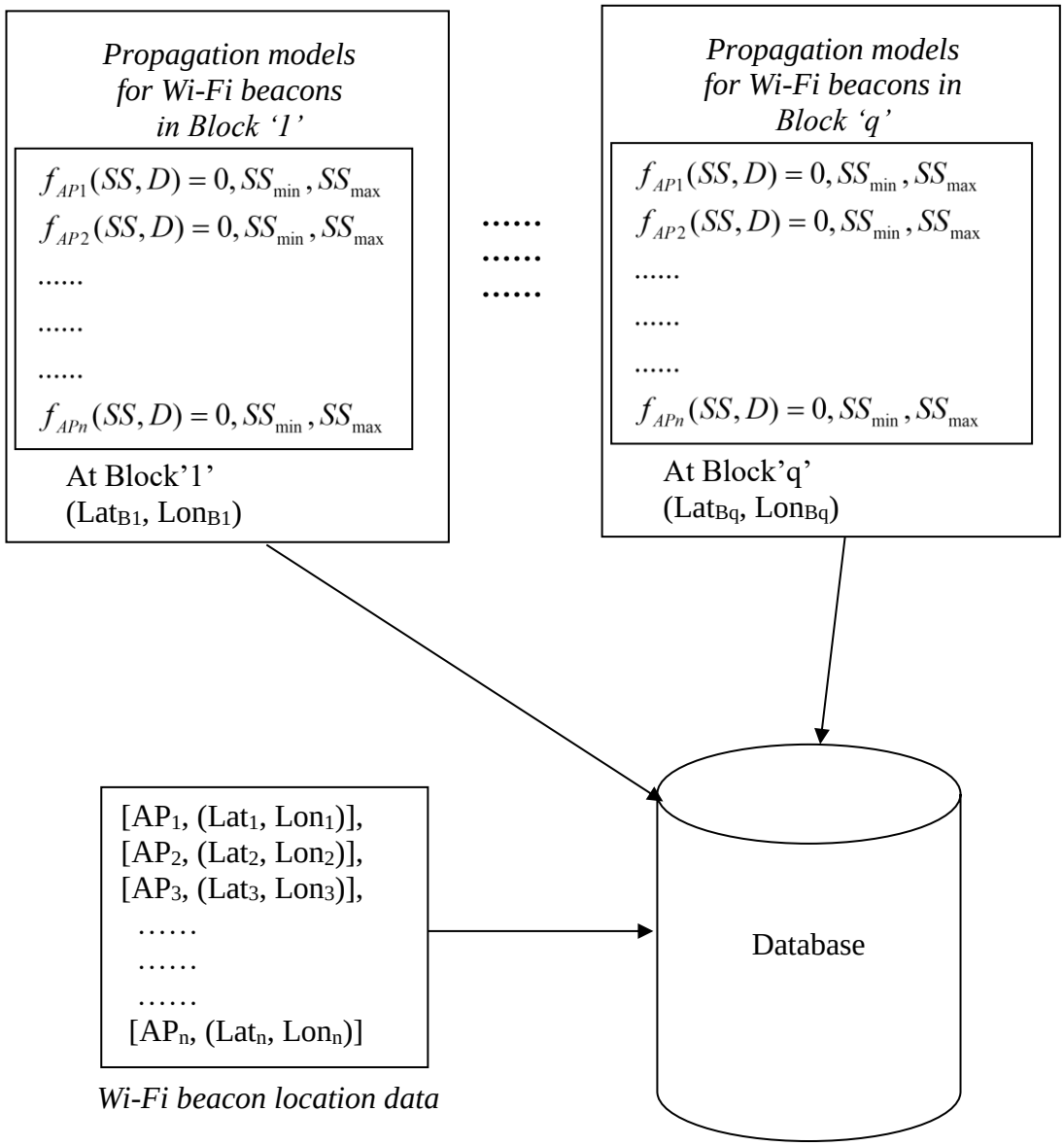


Figure 5.1 Updated database for an area of interest with 'q' blocks

5.2.1 Block identification

In the proposed positioning system before improvements, the block was identified by calculating Euclidean distance to stored calibration scans. In new system, the block is identified using only the propagation models. The block identification process is detailed in figure 5.2. The Euclidean distances calculated for each block using propagation models is referred to as 'Block distance'. Mobile device is estimated to be located in the block with smallest 'Block distance'.

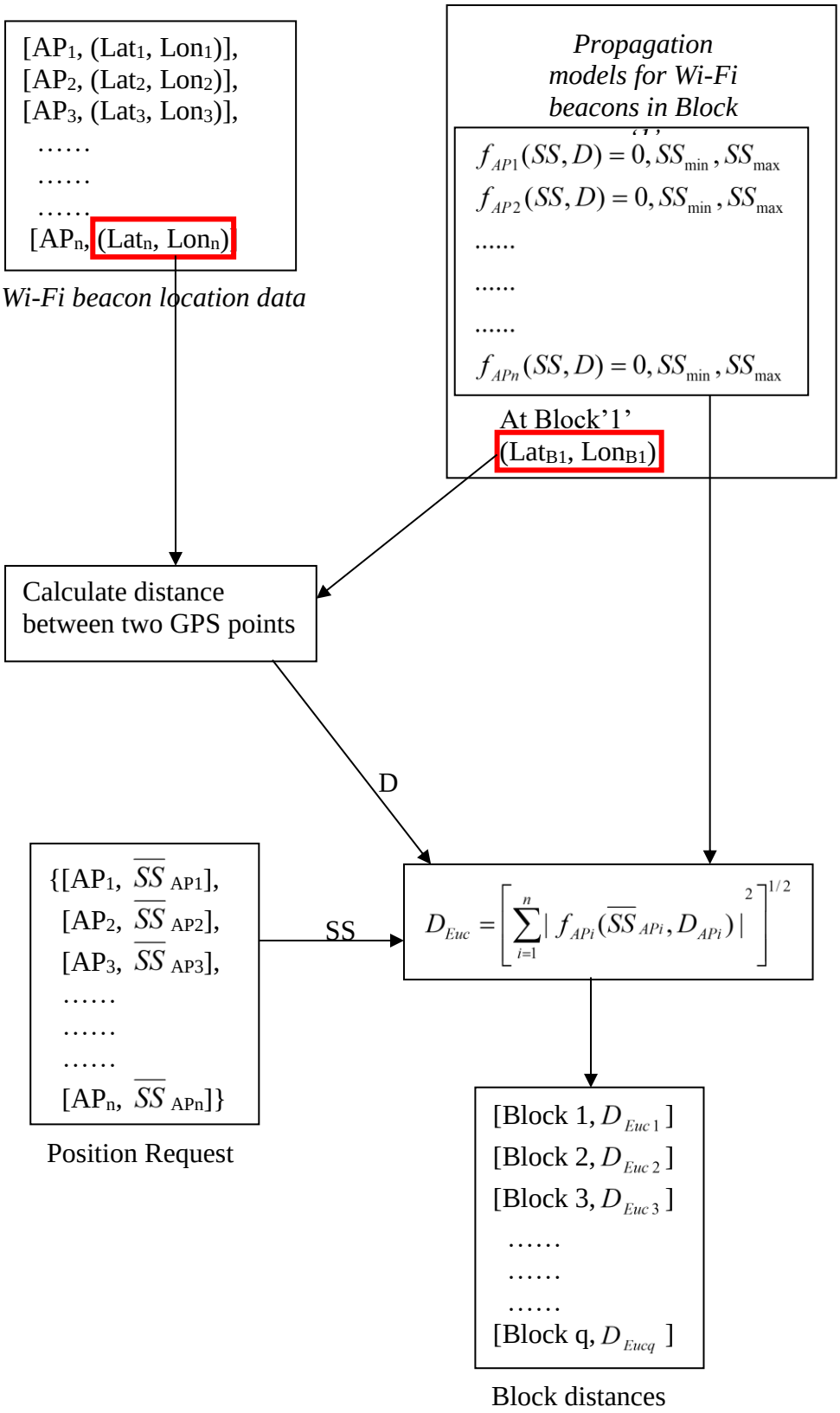


Figure 5.2 Block identification process

The block distance calculation process is detailed using a position request and data from database for block 1. The first step is to calculate distance (D) between centre of block and Wi-Fi beacons using equations (3.2) and (3.3). RSS of each Wi-Fi beacon from position request and distance (D) are substituted in corresponding propagation models ($f_{AP1}(SS, D) = 0$) of block 1. If RSS from position request are substituted in propagation models from the correct block, resulting value should be close to zero for each Wi-Fi beacon. For block 1, Euclidean distance is calculated using the following equation

$$D_{Euc} = \left[\sum_{i=1}^{n} | f_{APi}(\overline{SS}_{APi}, D_{APi}) |^2 \right]^{1/2} \quad \ldots\ldots \quad (5.1)$$

Euclidean distance is calculated for each block and the block which yields least value indicates location of mobile device.

The current block identification method still has to be robust enough to deal with missing Wi-Fi beacons in position request. The number of missing beacons in a position request can be up to 20% according to the analysis presented in previous chapter. The minimum and maximum RSS (SS_{min} and SS_{max}) of RSS values used in realizing the propagation model for each Wi-Fi beacon in each block are also stored as part of database as illustrated in figure 5.1. These values can be used as a means to decide the penalty that needs to be added for Wi-Fi beacons missing in the position request.

The previous fingerprint algorithm required storing of calibration scan collected at centre of the block, to be compared with position request during positioning phase. The current block identification method would replace the fingerprint algorithm in proposed positioning system earlier. This reduces data that needs to be stored to database.

5.3 Coordinate transformation for lateration Algorithm

The performance of current three-dimensional lateration algorithm was discussed towards the end of chapter 3. Also, the lateration algorithm was using local co-ordinate system which is not ideal for a system to be scalable. A scalable system would require lateration algorithm to use geographical coordinates. Since geographical (latitude-longitude) coordinates have uneven distance variation,

Geographical coordinates are transformed to Cartesian coordinates. The geographical coordinates can be transformed into three dimensional Cartesian coordinates with origin at centre of earth using transformation equations. The transformation between geographical and Cartesian coordinates has been widely studied over the years and a list of these studies can be found in [78].

5.3.1 Geographical coordinates and Cartesian coordinates

This section initially provides background to understand the transformation of geographical coordinates. Assigning coordinates to any point on the Earth is influenced by shape assumed to represent the Earth. The Earth can either be represented as a sphere or spheroid (ellipsoid) [79] shown in figure 5.3. However, the best representation of earth is spheroid.

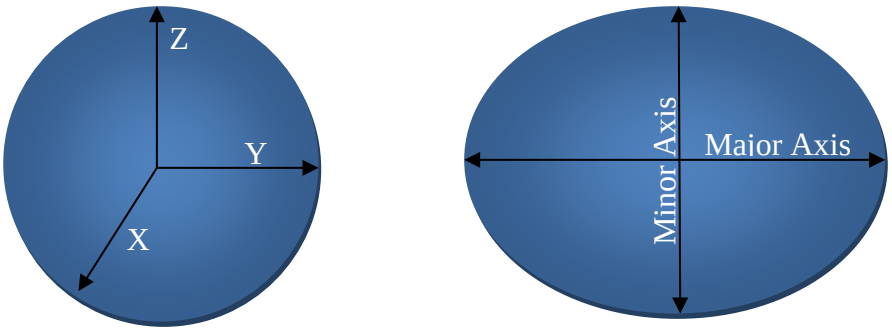


Figure 5.3 Approximations of shape of the Earth

The sphere is based on circle as shown in figure 5.3. Any point on surface of the earth is represented by X, Y and Z coordinates with origin at centre of the Earth. However, a spheroid is based on an ellipse which has a major and minor axis and any point on the Earth can also be represented by X, Y and Z coordinates with origin at centre of the Earth.

The most accurate representation of the shape of Earth is a spheroid. Three-dimensional Cartesian coordinates X, Y and Z, can be used to identify any point on surface of the Earth. Similarly, Latitude, Longitude and ellipsoid height can also be used to represent a point on surface of the Earth. The distribution of latitude and longitude on the surface of the earth are illustrated in figure 5.4 [80]. The coordinates are represented as ϕ (latitude), λ (longitude) and H (height). Latitudes (ϕ) are east-west lines which are also known as parallels. Longitudes (λ) are north-south lines

which are also known as meridians. Figure 5.4 shows a point ‘P’ on the surface of the Earth and corresponding coordinates in three dimensional Cartesian coordinates (X,Y,Z) and latitude-longitude and height coordinates (ϕ, λ, H).

.

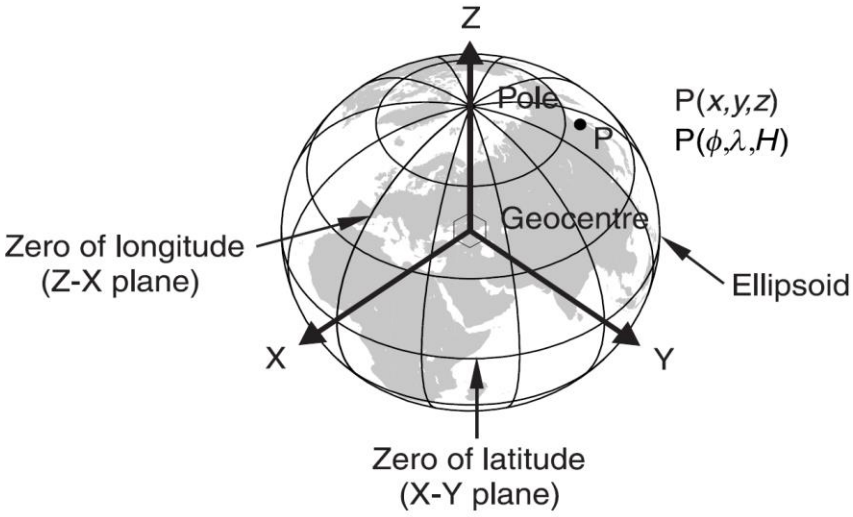


Figure 5.4 Spheroid showing Latitude, longitude and height and Cartesian coordinates

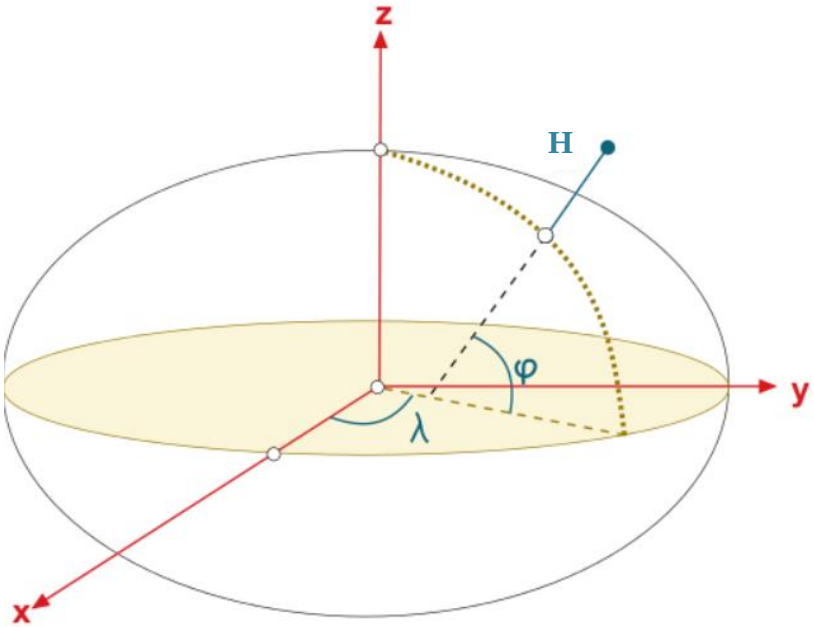


Figure 5.5 Illustration of geographical and Cartesian co-ordinate systems

Latitude (ϕ) of a point in figure 5.5 [81] is the angle between equatorial plane and the line perpendicular to spheroid at that point. Latitude values vary between 0 to

90 degrees north and 0 to 90 degrees south. But, the values for southern hemisphere can also be represented as negative values. Longitude (λ) of point 'P' is the angle between prime meridian and meridian passing through the point 'P'. Longitude values vary between 0 and 180 degrees east and 0 to 180 degrees west. The longitude values for western hemisphere are considered negative. The height 'H' is distance from surface of the spheroid.

Figure 5.5 also shows Cartesian co-ordinate system whose axes and origin are aligned with the latitude and longitude system. The X and Y axes line in the equatorial plane and Z axis is perpendicular to equatorial plane and passes through the poles.

5.3.2 Transformation between Geographical and Cartesian coordinates

The latitude-longitude co-ordinate system distances over the spheroid are not constant which makes computation with geographical coordinates complicated to run effectively on a mobile device. Consider an example, to compute distance between two coordinates, which was discussed in detail in section 3.2.1.2. Equations 3.2, 3.2 and 3.3 have to be evaluated to calculate distance between two geographical coordinates, while a simple equation 3.4 would have to be evaluated using cartesian coordinates. Lowering the computational intensity helps conserve power on a mobile device where available power is limited.

The following set of equations (5.2) describes transformation from latitude-longitude coordinates (ϕ, λ, H) to three-dimensional spheroid Cartesian coordinates (X, Y, Z).

$$\begin{aligned} X &= (N+H)\cos\phi * \sin\lambda \\ Y &= (N+H)\cos\phi * \cos\lambda \\ Z &= [(1-e^2) * N + H]\sin\phi \end{aligned} \qquad \ldots\ldots \qquad (5.2)$$

Where $N = \dfrac{a^2}{\sqrt{1-e^2\sin^2\phi}}$

Where 'e' is eccentricity given by $e = \sqrt{1-\dfrac{b^2}{a^2}}$

Where 'a' is semi-major axis

'b' is semi-minor axis

'H' is height from surface of the spheroid

The reverse transformation of three dimensional Cartesian coordinates to latitude-longitude coordinates can be performed using following equations (5.3) [82].

$$Longitude = \lambda = \tan^{-1}\left(\frac{Y}{X}\right)$$

$$Latitude = \phi = \tan^{-1}\left(\frac{Z}{\sqrt{X^2 + Y^2}}\left(1 + \frac{e^2 N \sin\phi}{Z}\right)\right) \quad \ldots\ldots(5.3)$$

$$Height = H = \frac{\sqrt{X2 + Y2}}{\cos\phi} - N$$

The calculation of latitude (ϕ) and height (H) are calculated iteratively and further details can be found in [82]. Some map projections which convert geographical coordinates to Cartesian coordinates can be found in [83].

5.3.3 Transformation between Geographical and local 2D-coordinates

The previous section discussed transformation of coordinates between latitude-longitude coordinates system to three Cartesian dimensional coordinates system. The drawbacks of three dimensional Cartesian coordinates as discussed in the previous chapter are

1) The lateration algorithm would still require at least four spheres for solving the location of mobile device
2) The transformation from three dimensional coordinates to latitude-longitude coordinates is an iterative process, hence computationally intensive
3) The least squares algorithm for lateration is also an iterative process which updates the current result based on residuals using previous result, adding to computational intensity

This section introduces a few terms and formulae used in proposed algorithm in following section. The calculations using latitude-longitude coordinates assume the Earth to be a sphere.

5.3.3.1 Distance calculation between two Geographical coordinates

The shortest distance on the surface of sphere is known as Great circle distance [84] [85] or Orthodromic distance. The distance between two points on earth

represented by latitude-longitude (geographical) coordinates is calculated using Haversine formula [86]. Consider two points (A and B) on surface of the earth and let the coordinates of A = (ϕ_1, λ_1) and B = (ϕ_2, λ_2). The mathematical formulae for distance between points 'A' and 'B' are listed as equation (5.4).

$$\text{Distance} = d = R * c \quad \ldots\ldots \quad (5.4)$$

Where 'R' is the radius of earth (6371km)

$$c = 2 * \tan 2^{-1}\left(\sqrt{a}, \sqrt{1-a}\right)$$

$$a = \sin^2\left(\frac{\nabla\phi}{2}\right) + \cos\phi_1 * \cos\phi_2 * \sin^2\left(\frac{\nabla\lambda}{2}\right)$$

$$\nabla\phi = \phi_2 - \phi_1$$

$$\nabla\lambda = \lambda_2 - \lambda_1$$

5.3.3.2 Bearing calculation

Bearing is the angle between geographical north and line joining the current point and a distant point. This can be explained by considering a point 'A' with coordinates (ϕ_1, λ_1) and a point 'B' with coordinates (ϕ_2, λ_2) as illustrated in *figure 5.6*. The bearing angle from point 'A' to point 'B' is represented as 'Θ_{AB}'.

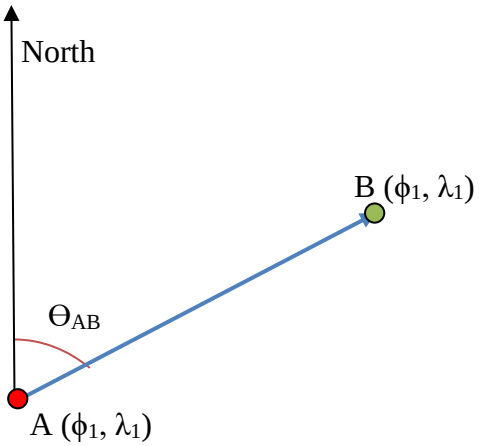


Figure 5.6 Bearing from point 'A' to point 'B'

Mathematical formula for calculating the bearing angle 'Θ_{AB}' ([85], [87] and [88]) is

$$\theta = \tan 2^{-1}(\sin\Delta\lambda * \cos\phi_2, \cos\phi_1 * \sin\varphi_2 - \sin\phi_1 * \cos\phi_2 * \cos\Delta\lambda) \ldots\ldots (5.5)$$

Where $\Delta\lambda = \lambda_2 - \lambda_1$

Since Atan2 (also written as $\tan 2^{-1}$), returns values between -180 to +180 to normalise the value, 360 degrees is added to Θ and modulus of the sum gives bearing,

$$\theta_{AB} = (\theta + 360)\%360 \quad \ldots\ldots \quad (5.6)$$

In equation 5.6, '%' represents modulo operation, which returns the remainder after division by 360. The resulting bearing 'Θ_{AB}' ranges between 0 to 360 degrees.

5.3.3.3 Cross track distance calculation

Cross track distance is the error distance between intended track and current location. To explain cross track distance further, consider the scenario illustrated in *figure 5.7*. The intended track is from point 'A' to point 'B' but the current position is at point 'C'.

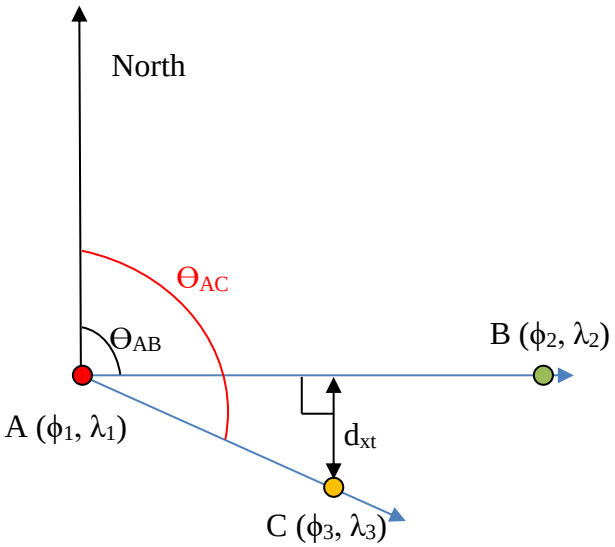


Figure 5.7 Cross track distance

In figure 5.7, cross track distance is represented as 'd_{xt}', the bearing from point 'A' to point 'B' is 'Θ_{AB}' and bearing from point 'A' to point 'C' is represented as 'Θ_{AC}'. Track from 'A' to be 'B' represents the intended track and the current position is at point 'C'. Cross track distance 'd_{xt}' is the perpendicular distance from point 'C' to track 'AB'. The formula to calculate cross track distance is ([87] and [88])

$$d_{xt} = \sin^{-1}\left(\sin\left(\frac{d_{AC}}{R}\right) * \sin(\theta_{AC} - \theta_{AB})\right) * R \quad \ldots\ldots \quad (5.7)$$

Where d_{AC} is distance from point A to C and can be calculated using equation (5.4)

R is the radius of earth (6371Km)

5.3.3.4 Calculating coordinates of a point given bearing and distance

This section discusses a scenario illustrated in figure 5.8, where the coordinates of point 'A' and bearing to point 'B' are known. The coordinates of point 'B' along the great circle at a distance 'd_{AB}' have to be calculated. The formulae to calculate coordinates of point 'B' are as follows ([87] and [88]).

$$\phi_2 = \sin -1\left(\sin\phi_1 * \cos\left(\frac{d_{AB}}{R}\right) + \cos\phi_1 * \sin\left(\frac{d_{AB}}{R}\right) * \cos\theta_{AB}\right)$$
$$\lambda_2 = \lambda_1 + \tan 2^{-1}\left(\sin\theta_{AB} * \sin\left(\frac{d_{AB}}{R}\right) * \cos\phi_1, \cos\left(\frac{d_{AB}}{R}\right) - \sin\phi_1 * \sin\phi_2\right) \quad .. (5.8)$$

Where R is the radius of earth (6371Km)

d_{AB} is the distance between points 'A' and 'B'

Θ_{AB} is the bearing from point 'A' to point 'B'

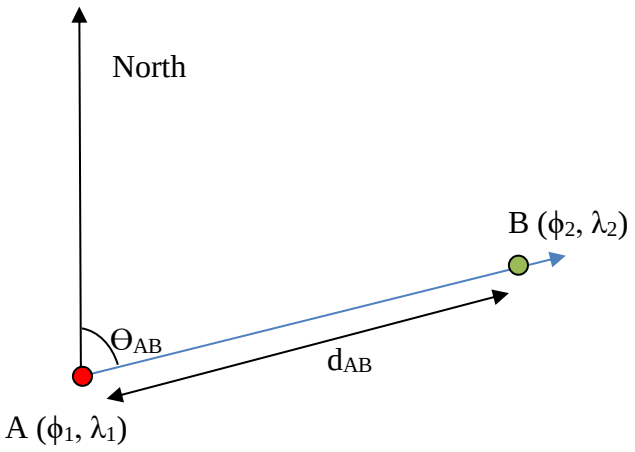


Figure 5.8 Calculation of coordinates given distance and bearing

Equations (5.8) calculate latitude (ϕ_2) and longitude (λ_2) of point 'B' which is at a distance 'd_{AB}' from point 'A' and bearing 'Θ_{AB}'.

5.3.3.5 Transformation algorithm

This section describes an algorithm to transform geographical coordinates (latitude-longitude) into a two-dimensional local co-ordinate system. The proposed algorithm uses the formulae (5.4), (5.6), (5.7) and (5.8) detailed in the previous sections. Figure 5.9 shows the latitude-longitude coordinates and corresponding local X-Y coordinates along with a test area highlighted in blue.

5.3.3.5.1 Geographical coordinates to local 2D-coordinates

The first step is to select an origin (O) such that origin either aligns or beyond southernmost and westernmost points of the test area. This allows resulting two dimensional coordinates to be all positive as they lie in the first quadrant.

The second step is to select end of axis represented in the figure 5.9 as 'X' and 'Y' which are ends of the axis. The point 'Y' is calculated such that the bearing from origin is zero degrees and be so far north as to cover the test area. Similarly, the point 'X' is selected such that the bearing from origin is ninety degrees and so far, east as to cover the test area.

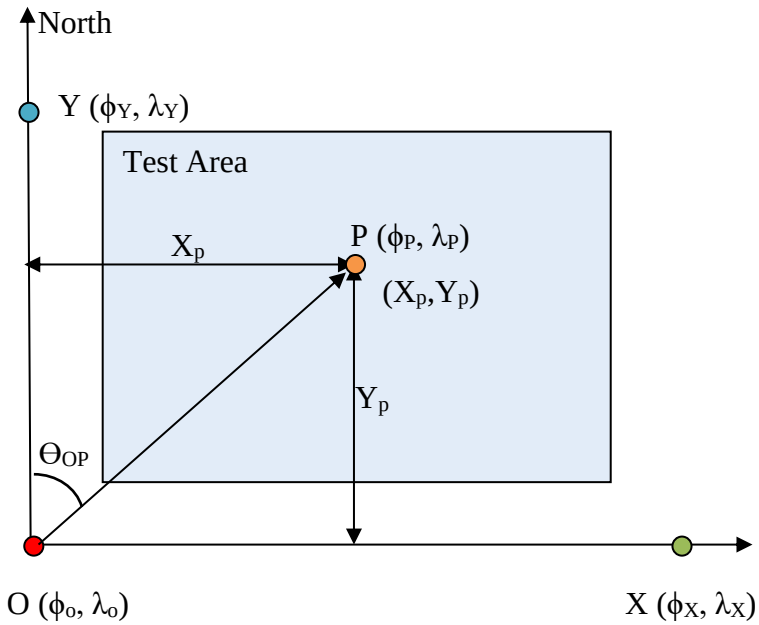


Figure 5.9 Two-dimensional local co-ordinate system

The steps detailed above create reference X-Y axis, based on which, the two-dimensional local coordinate system is built. The local coordinates of point 'P' in figure 4.8 are 'X_p' and 'Y_p'. 'X_p'is the cross-track distance from track 'OY', which can be calculated using the equation (5.7). Similarly, 'Y_p' is the cross-track distance from track 'OX'. Any points in test area can be transformed into two-dimensional local coordinates using same technique.

5.3.3.5.2 Local 2D-coordinates to Geographical coordinates

This section details the steps to convert the two-dimensional coordinates to latitude-longitude coordinates. The technique of calculating coordinates of a point

using the bearing and distance from a known point is used for reverse transformation. For reverse transformation, in figure 5.9, consider the coordinates (X, Y) of 'P' in local coordinate system are known and 'O' is the origin. The distance between the origin and point 'P' can be calculated using Pythagoras theorem ($d_{OP} = \sqrt{X^2 + Y^2}$). The bearing from origin 'O' to point 'P' is calculated using basic trigonometry where $\theta_{OP} = \tan^{-1}\left(\frac{X}{Y}\right)$. The distance and bearing angle from point 'O' and 'P' are now known and since the latitude-longitude coordinates of point 'O' are also known. Using the equations (5.8) the latitude-longitude co-ordinates of point 'P' can be calculated.

5.4 Performance of Latertion algorithm

This section compares implementations of Lateration, linearization algorithm and Non-Linear least squares algorithms. The implementations transformed three dimensional Cartesian coordinates and two dimensional local coordinates. The linearization and least square algorithms were discussed earlier in chapter 2. The implementations of these algorithms were done in Matlab. The implementations were used to calculate the location of Wi-Fi beacon given approximate distances to the beacon from multiple known locations. The accuracy of calculated beacon location reflects performance of the implementations.

5.4.1 Test area and data collection

The data used to test lateration implementation was collected in Barcelona at MWC (Mobile World Congress). The data required for the testing the current implementations were, approximate distances to Wi-Fi beacons from multiple locations and latitude-longitude coordinates of the locations. The actual location coordinates of Wi-Fi beacons were also required to calculate accuracy of estimated location. The data required for the current tests was provided by "sensewhere" which was collected by the company when attending MWC. Linearization and non-linear least squares implementations were applied to estimate the location of Wi-Fi beacons. The estimated locations were compared with actual locations to evaluate the performance of linearization and non-linear least square algorithms.

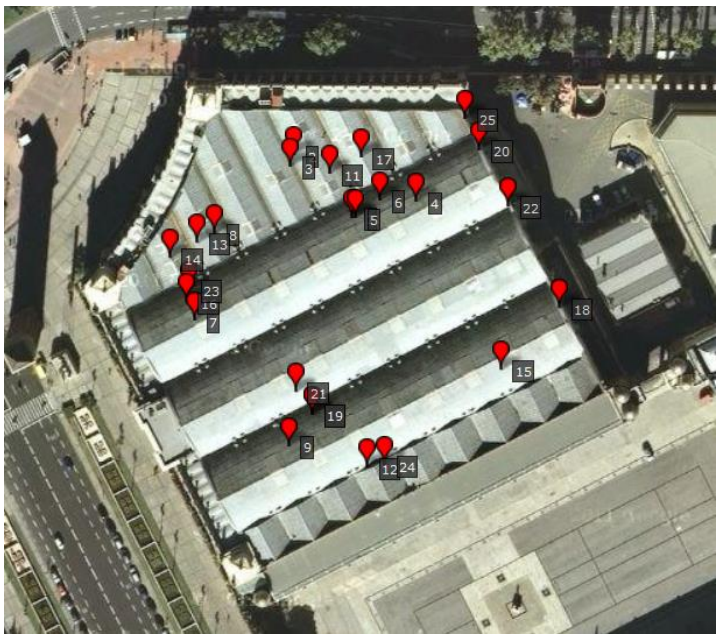


Figure 5.10 Wi-Fi beacon locations in Hall 1

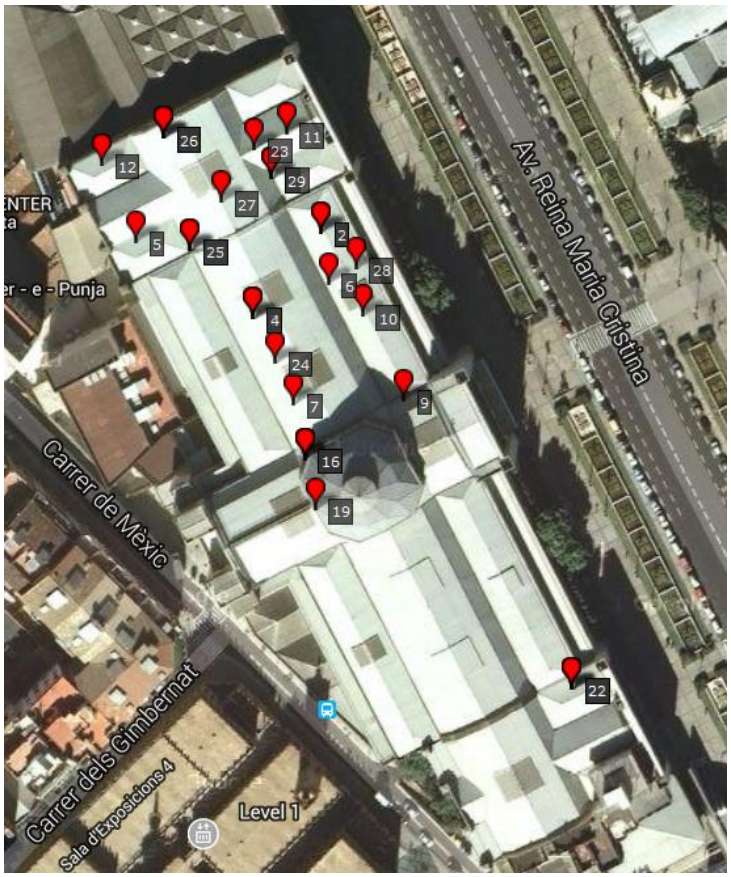


Figure 5.11 Wi-Fi beacon locations in Hall 8

The actual location of Wi-Fi beacons is shown in figures 5.10 and 5.11. There are 54 Wi-Fi beacons shown and the data was collected at two of the halls (1 and 8) at Mobile World Congress (MWC) venue.

5.4.2 Lateration algorithms with three dimensional coordinates

The two implementations of Lateration in Matlab used linearization algorithm and non-linear least squares algorithms. A total of 54 lateration cases were calculated of which 25 were from Hall 1 and 29 from Hall 8. The implemented code first transforms the latitude-longitude coordinates to three dimensional Cartesian coordinates and then lateration algorithms are applied. Figure 5.12 shows accuracy results for 25 test cases from Hall 1 using lateration algorithm. The 90th percentile accuracy with linearization algorithm was 31m and 23m with non-linear least squares. This shows an improvement of 25.8% using non-linear least squares compared to linearization algorithm.

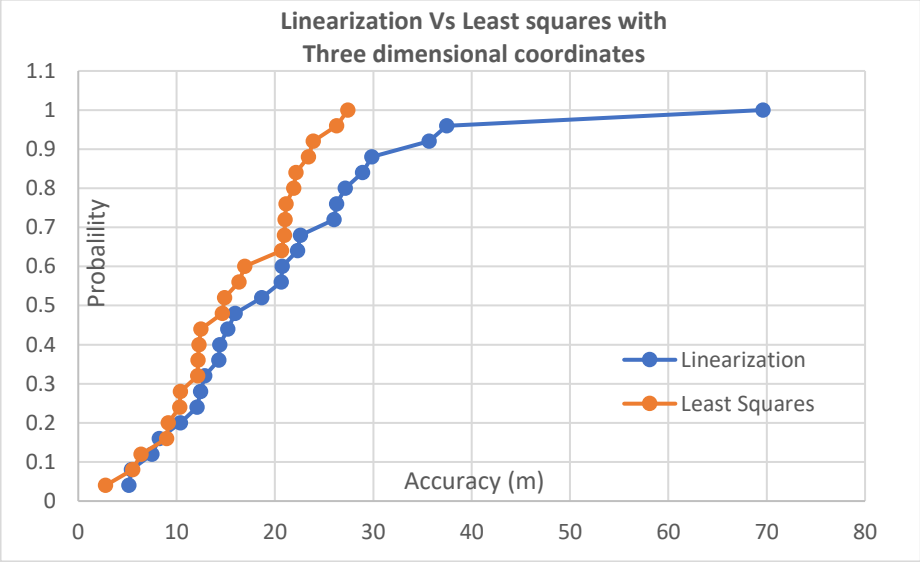


Figure 5.12 Accuracy comparison - Linearization Vs Least squares with three dimensional coordinates (Hall 1)

The lateration process was repeated on 29 test cases from Hall 8 and results are illustrated in figure 5.13. The 90th percentile accuracy using linearization algorithm is 30m and 20m using non-linear least squares algorithm which is an improvement of 33.3%.

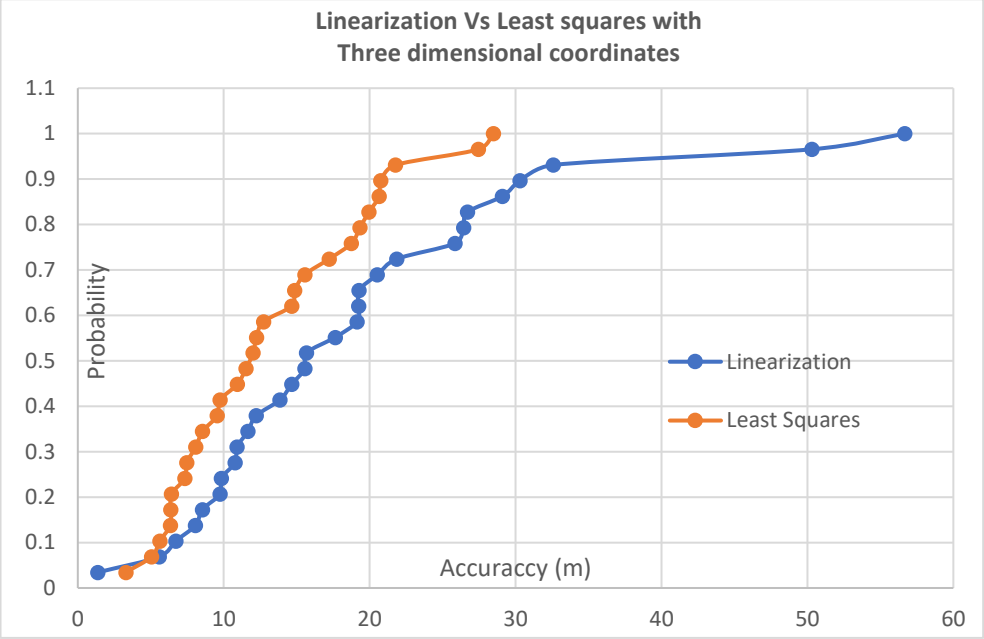


Figure 5.13 Accuracy comparison - Linearization Vs least squares with three dimensioanl coordinates (Hall 8)

5.3.3 Lateration algorithms with two dimensional coordinates

Linearization algorithm and non-linear least squares algorithm were implemented in Matlab using two dimensional local coordinates. A total of 54 lateration cases were calculated from Hall 1 and 8. The implemented code first transforms the latitude-longitude coordinates to two dimensional local Cartesian coordinates and then lateration algorithms are applied. Figures 5.14 and 5.15 show accuracy results for same test cases from Halls 1 and 8 discussed in previous section. The 90th percentile accuracy with linearization algorithm was 30m and 23m with non-linear least squares for test cases from Hall 1 as illustrated in figure 5.14. This shows an improvement of 23.3% using non-linear least squares compared to linearization algorithm.

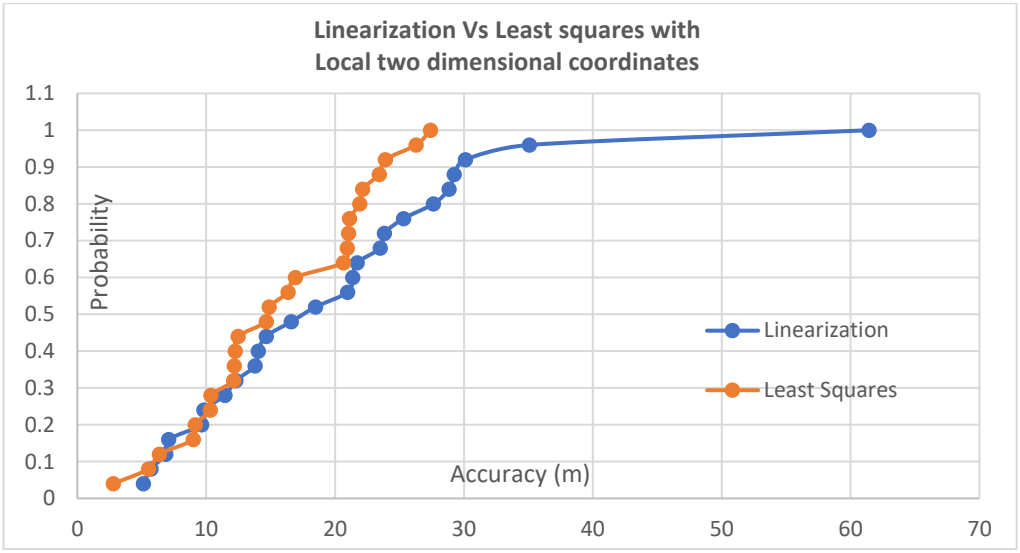


Figure 5.14 Accuracy comparison - Linearization Vs Least squares with two dimensional coordinates (Hall 1)

Figure 5.15 shows accuracy results for Hall 8 using linearization and non-linear least squares algorithms. The 90th percentile accuracy with linearization algorithm was 31m and 20m with non-linear least squares, which is about 35.4% better performance.

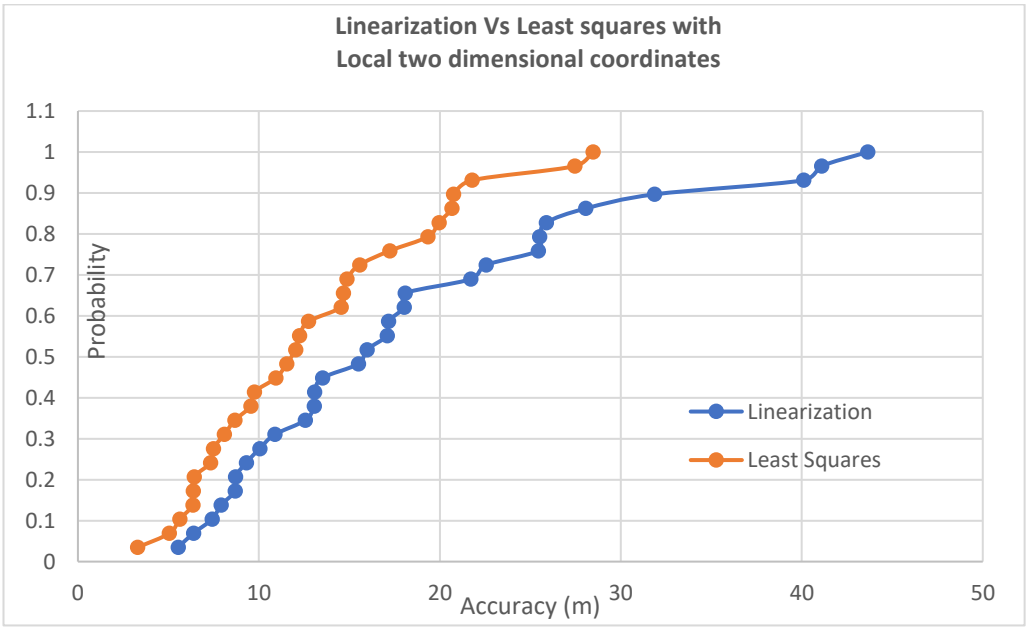


Figure 5.15 Accuracy comparison - Linearization Vs Least squares with two dimensional coordinates (Hall 8)

5.5 Analysis of Lateration algorithm

The accuracy of linearization algorithm and non-linear least squares algorithm were discussed in previous section. Table 5.1 summarizes the performances for test cases in Hall 1 and Hall 8.

Accuracy	Hall 1 (25 test cases)		Hall 8 (29 test cases)	
	3d (m)	2d (m)	3d (m)	2d (m)
Linearization algorithm	31	30	31	30
Non-linear least squares algorithm	23	23	30	31

Table 5.1 Accuracy of test cases under multiple scenarios

Lateration using three dimensions includes ‘height’. In case of three dimensional co-ordinates, ‘Z’ coordinate is measured from the centre of the earth. The accuracy for lateration test cases in table 5.1 with three dimensional coordinates did not include the height. The accuracy was calculated as distance between two latitude-longitude coordinates. The average error in height estimation for test cases in Hall 1 using linearization algorithm was 376.8m. But using non-linear least squares algorithm, the average error in height estimation was less than 0.5m as illustrated in figure 5.16. Similar results were observed for test cases in Hall 8, where the average height accuracy using linearization algorithm was 340.3m whereas with non-linear least squares algorithm, it was 0.6m as illustrated in figure 5.17.

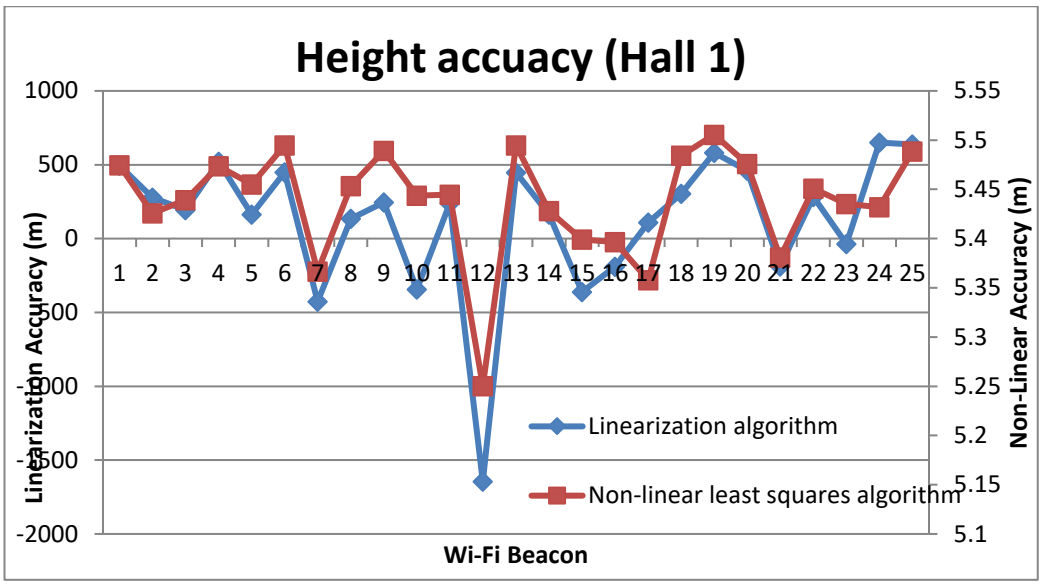


Figure 5.16 Height accuracy using three dimensional coordinates (Hall 1)

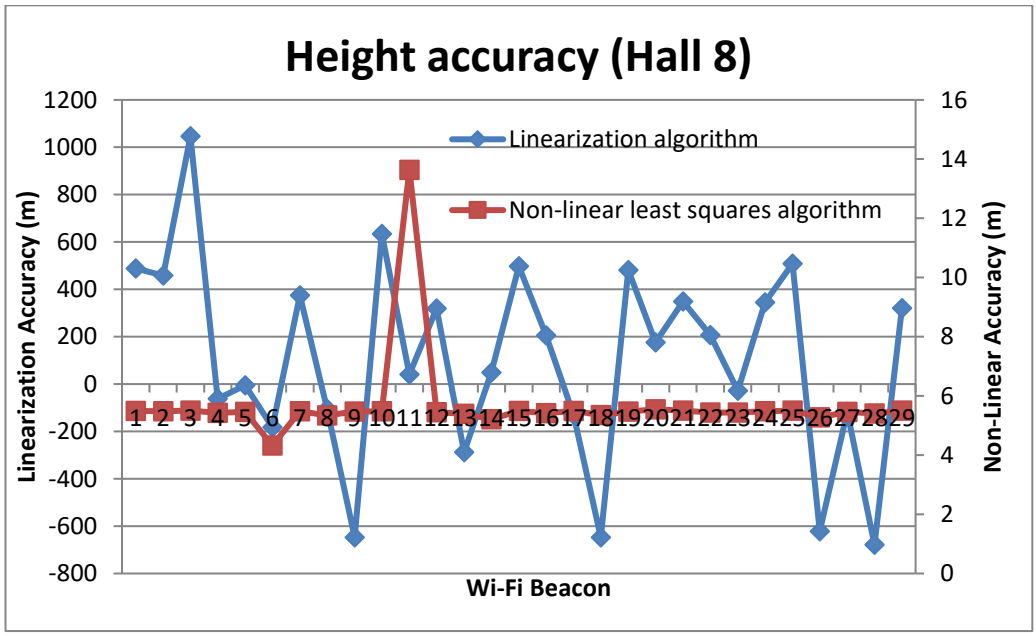


Figure 5.17 Height accuracy using three dimensional coordinates (Hall 8)

It is clear that using linearization algorithm, height estimation has particularly large errors. Since the non-linear least squares algorithm is an iterative process, the height estimation has better accuracy.

Another important aspect to be analysed with non-linear least squares algorithm is the number of iterations it takes to reach the solution. The number of iteration for lateration at test cases in Hall 1 and Hall 8, for both three and two-dimensional least squares were noted. The average number of iterations for 25 test

cases in Hall 1 using three-dimensional least squares was 30.2 and using two dimensional least squares the number of iterations reduced to 12.8 which is about 57% fewer iterations. The number of iterations it took to reach the solution in each test case is illustrated in figure 5.18.

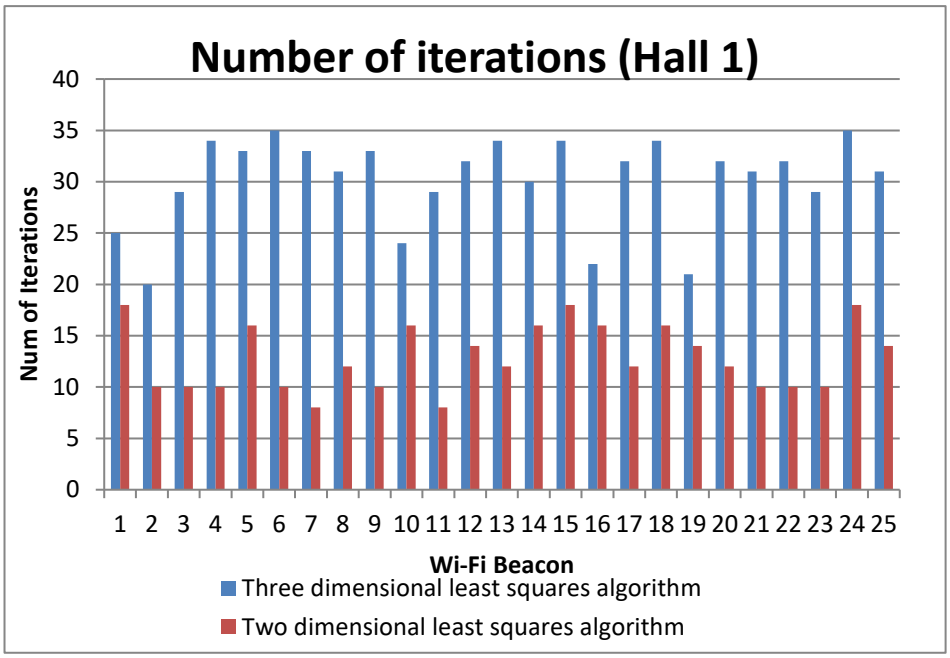


Figure 5.18 Number of iterations for test cases in Hall 1

The number of iterations required for test cases in Hall 8 are shown in figure 5.19, which also shows, about 57% fewer iterations were required when using two dimensioanl least squares.

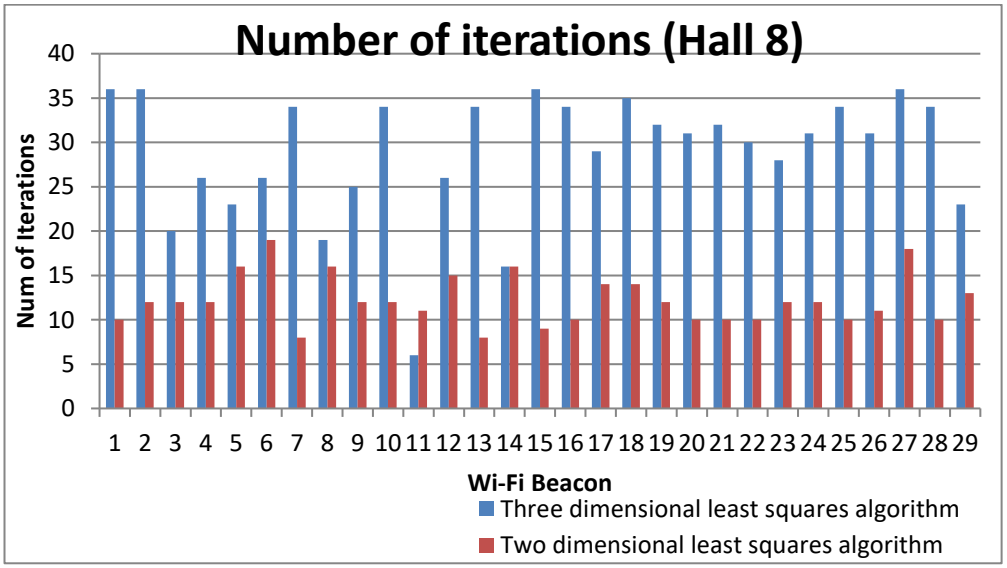


Figure 5.19 Number of iterations for test cases in Hall 8

5.6 Introduction to implementation of improvements

The experimental implementation of the proposed improvements to system is detailed in this chapter. The test area is Sanderson building at King's Buildings campus of University of Edinburgh which was used for testing the initial proposed system. The floor plans for the building were obtained from the university and Figure 5.20 shows overlay of floor plan on Google maps. The current implementation uses only the first floor of the building and test area is highlighted.

Figure 5.20 Test area on first floor of Sanderson building

5.7 Calibration phase

5.7.1 Test area and Calibration points

The test area on first floor of Sanderson building is highlighted in *figure 5.20.* The test area was divided into 11 blocks as shown in *figure 5.21*.

Figure 5.21 Test area divided into blocks

The test area consists of two corridor areas and five classrooms. The long corridor area was divided into six blocks (B1 to B6), each with dimensions 10m×2.5m. The second corridor area was approximately 10m×8m which is highlighted as Block 8. Rest of the test area are classrooms which are identified as individual blocks (B7, B9, B10 and B11).

Calibration points were selected such that they were approximately two meters apart. A total of 174 calibration points were selected and these points are shown in *figure 5.22*.

Figure 5.22 Calibration points

5.7.2 Wi-Fi beacon location data collection

The Wi-Fi beacons location was collected by surveying the test area. Six beacons were identified and location coordinates were obtained using floor plans. Each physical Wi-Fi beacon on the university network transmits four MAC ids. For example, Beacon 1 (shown in *figure 5.23*) transmits four MAC ids 00:13:5F:F8:73:20, 00:13:5F:F8:73:21, 00:13:5F:F8:73:2E and 00:13:5F:F8:73:2F. In the current implementation each MAC id is considered as an individual beacon. The current test area has six beacons but Wi-Fi beacon location data has a list of 24 beacons and their location. The locations of the six Wi-Fi beacons are displayed in *figure 5.23*.

Figure 5.23 Location of Wi-Fi beacons in test area

5.7.3 Data collection at calibration points

The calibration data was collected using an android application from company called ‘sensewhere’. A screenshot of the application is shown in figure 5.24. The application shows the internal floor plan of Sanderson building overlaid on Google maps. The location where manual data is to be collected is indicated by pressing on the map. A red icon shows selected location as illustrated in figure 5.24. The location can be adjusted using the arrows displayed on the screen. The scan data is collected by touching the ‘seed’ button at the bottom right corner (highlighted in blue on figure 5.24). This process is repeated at the same location facing north, east, south and west directions. The application tags data collected at each calibration point with latitude-longitude coordinates.

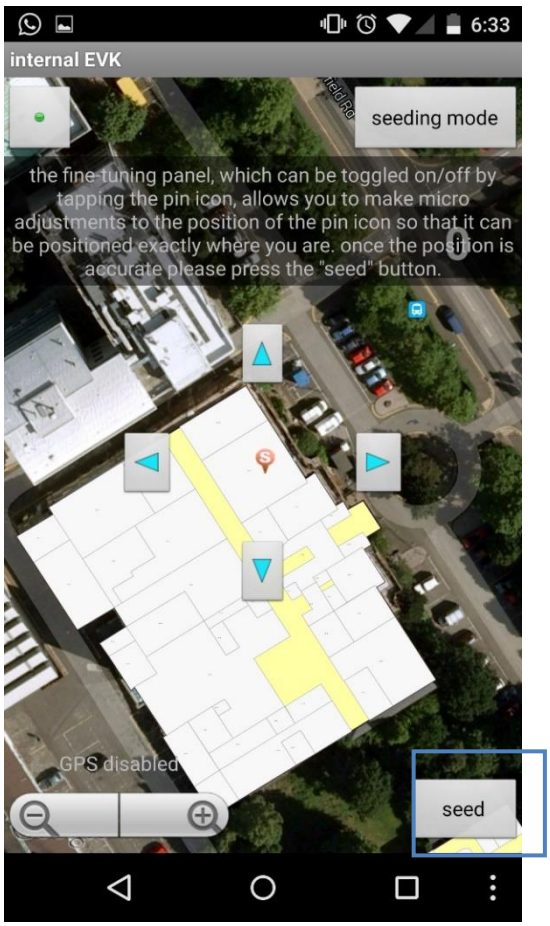


Figure 5.24 Screenshot of sensewhere application

5.7.4 Data processing

The data collected at each calibration point is processed to average the RSS values obtained facing various directions. The resulting scan data at each calibration point is as shown in figure 5.25.

MAC id and corresponding average RSS

00:1B:2B:6B:18:61	-84	55.92326	-3.17208
00:1B:8F:88:A0:B0	-86.25	55.92326	-3.17208
00:1B:8F:88:A0:B1	-86	55.92326	-3.17208
00:1E:F7:EB:20:00	-81.25	55.92326	-3.17208
00:1E:F7:EB:20:01	-80.5	55.92326	-3.17208
00:1E:F7:EB:20:0E	-91	55.92326	-3.17208
00:1E:F7:EB:20:0F	-92	55.92326	-3.17208
34:A8:4E:FD:68:E0	-83.6667	55.92326	-3.17208
34:A8:4E:FD:68:E1	-84	55.92326	-3.17208
54:78:1A:21:3B:C0	-63.5	55.92326	-3.17208
54:78:1A:21:3B:C1	-63	55.92326	-3.17208

Latitude –Longitude coordinates of calibration point

Figure 5.25 Processed data at a calibration point

The processed data for each calibration point is used along with the Wi-Fi beacon location data to estimate distances between calibration point and Wi-Fi beacon. This process is similar to the processing described in chapter 3. The coordinates used in chapter 3 were local three-dimensional coordinates where as in the current implementation latitude-longitude coordinates are used. The distance between the calibration point and Wi-Fi beacon is calculated using equation 5.4 which gives the distance between two points with latitude-longitude coordinates. After calculation of distances between calibration point and Wi-Fi beacons, this data is used to build propagation models for each Wi-Fi beacon in each block (similar to process described in chapter 3).

The propagation models for each Wi-Fi beacon in a block are realised. Figure 5.26 shows processed data for Block 1 which consists of MAC id, propagation model, Block coordinates, minimum RSS and maximum RSS.

34:A8:4E:FD:68:E0	y = -36.471x - 34.945	55.92326	-3.17208	-74	-86
34:A8:4E:FD:68:E1	y = -32.379x - 40.581	55.92326	-3.17208	-75.25	-86.6667
34:A8:4E:FD:68:EE	y = -52.981x - 23.867	55.92326	-3.17208	-82.5	-93
34:A8:4E:FD:68:EF	y = -43.244x - 37.245	55.92326	-3.17208	-82	-96
54:78:1A:21:3B:C0	y = 7.5845x - 73.951	55.92326	-3.17208	-60.25	-73.75
54:78:1A:21:3B:C1	y = 6.6349x - 72.953	55.92326	-3.17208	-60.5	-73
54:78:1A:21:3B:CE	y = -0.8805x - 75.547	55.92326	-3.17208	-71.5	-82.25
54:78:1A:21:3B:CF	y = 4.2047x - 79.665	55.92326	-3.17208	-71	-80.75

Figure 5.26 Stored data for Block 1 after processing

A fingerprint system would require storing data from 174 calibration points. The size of data when stored to a file, from 174 calibration points was 237.95KB. Following the data processing required for the algorithm being implemented, the data stored was 10.01KB. This is approximately 23 times smaller compared to a fingerprint algorithm.

5.8 Positioning Phase

During positioning phase, position request data from the mobile device is processed along with database generated during calibration phase to estimate the location of mobile device. 26 test points were identified in the test area and location of test points are shown in figure 5.27. Positioning phase involves, first identifying the bock in which mobile device is located and then using propagation models from corresponding block to estimate the distances to Wi-Fi beacons. The estimated

distances are then used as in lateration algorithm to refine the estimated location further. The following sections detail the accuracy results for block identification and lateration processes.

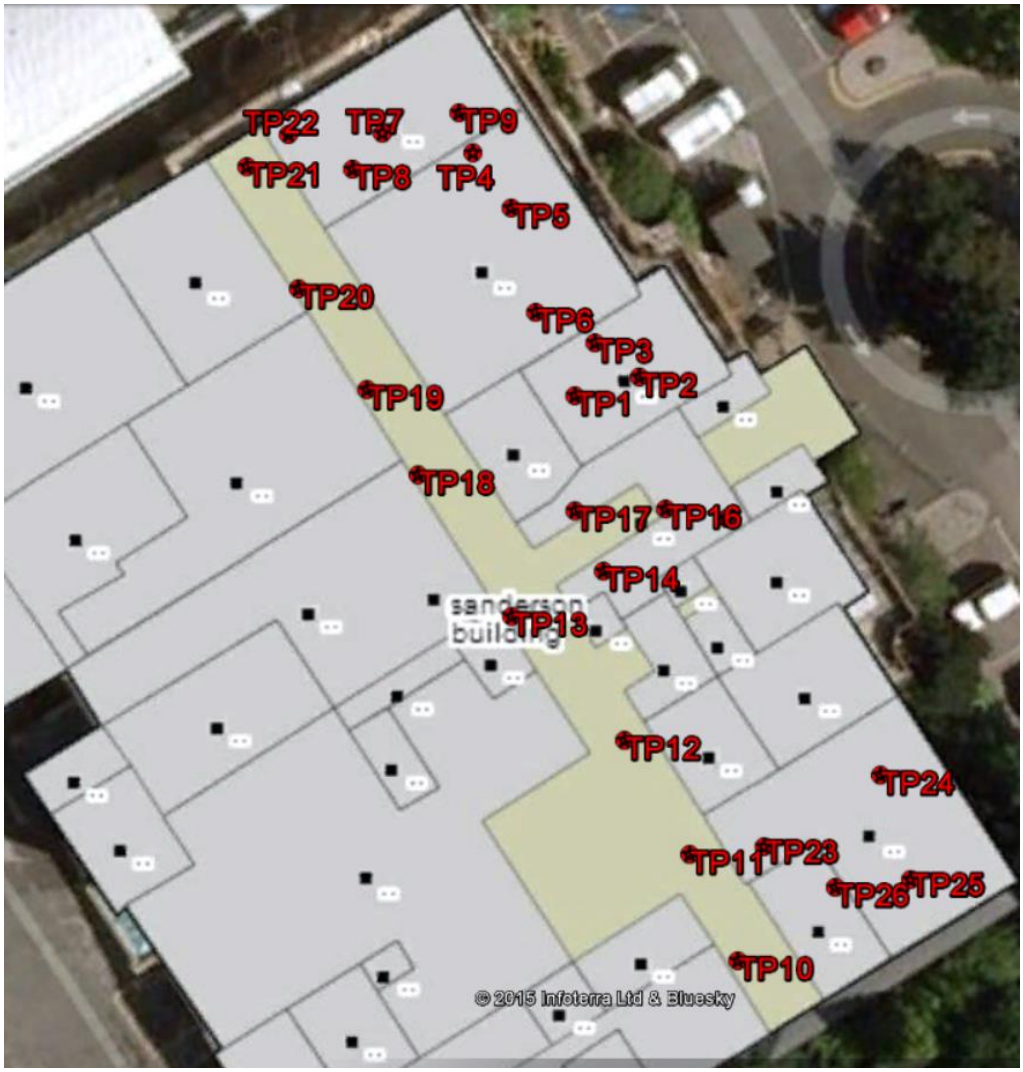


Figure 5.27 Location of test points

5.8.1 Block identification results

The test points are spread over 11 blocks as shown in figure 5.27. The block distance is calculated for each block using the RSS values from position requests. The block distance calculation is detailed in block diagram shown in figure 5.2. The distance from the centre of block to Wi-Fi beacon and corresponding RSS from position request are substituted in the propagation model. A propagation model from the correct block results in least residual. The residual is calculated for all Wi-Fi beacons in the block, and for all 11 blocks. The block with minimum residual value (block distance) is identified as the block in which mobile device is located. Block

identification results for 26 test points are as shown in table 5.2. The test points for which wrong block was identified are highlighted. The accuracy of block identification for 26 test points is 84.6%.

Test Point	Actual Block	Estimated Block
TP1	B9	B9
TP2	B9	B9
TP3	B9	B9
TP4	B10	B10
TP5	B10	B10
TP6	B10	B10
TP7	B11	B11
TP8	B11	B1
TP9	B11	B11
TP10	B6	B5
TP11	B6	B6
TP12	B5	B5
TP13	B4	B3
TP14	B8	B8
TP15	B8	B8
TP16	B8	B8
TP17	B8	B3
TP18	B3	B3
TP19	B2	B2
TP20	B1	B1
TP21	B1	B1
TP22	B11	B11
TP23	B7	B7
TP24	B7	B7
TP25	B7	B7
TP26	B7	B7

Table 5.2 Block identification results

For the current test area, block identification requires 11, block distance computations for each test point. In case of a fingerprint system, 174 signal distance calculations are required for test area with 174 calibration points. The proposed system requires about 15 times less calculations compared to a fingerprint system.

5.8.2 Lateration results

After estimating the block in which the mobile device is located, propagation models corresponding to the block were used to estimate distances to Wi-Fi beacons. The coordinates of Wi-Fi beacon location, were transformed to local two-dimensional coordinates for lateration. The results of lateration process using two dimensional coordinates are shown in figure 5.28.

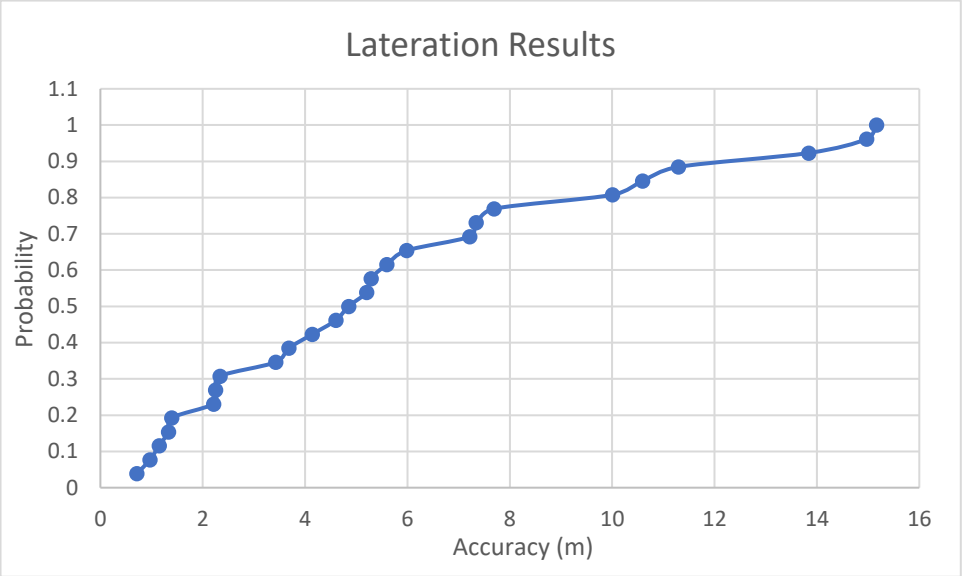


Figure 5.28 Lateration results

The 90th percentile accuracy for 26 test points is 11m. Some of the test points had errors over to 10m. The reasons for these large errors are discussed in the following sections.

5.9 Fingerprinting system results

Nearest neighbour fingerprinting algorithm implementation was applied to the same data collected at Sanderson building. The results at the same 26 test points are shown in figure 5.29. The 90th percentile accuracy for the test points shown in figure 5.10 is 6.3m and 50th percentile accuracy is 2.8m. Fingerprinting results in [117] and [118] also show 50th percentile accuracies of close to 3m.

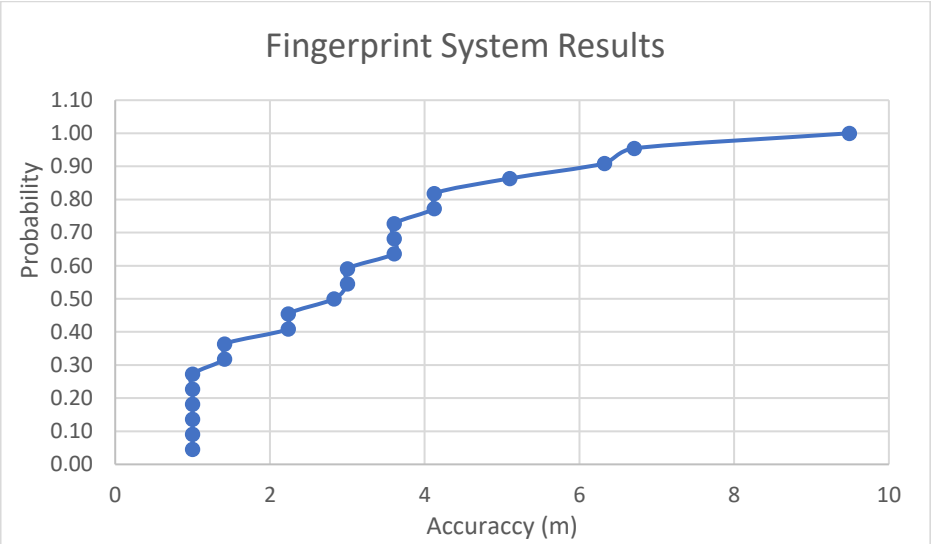


Figure 5.29 Fingerprint system results

5.10 Analysis of Results after Improvements implementation

This chapter investigates into the reasons for errors at some test points. At most of the test points, lateration helped to refine the location within an identified block. However, at some test points, where identified block was wrong, meant the propagation models used in distance estimation resulted in large errors. In the following sections, a test point is used to describe block identification process and then discusses reasons for errors at some test points.

5.10.1 Block identification

Block identification process for test point 1 is detailed in this section. The position request at test point is shown in table 5.3.

MAC Id	RSS
00:23:5E:95:E4:E1	-84
34:A8:4E:FD:68:E0	-56
34:A8:4E:FD:68:E1	-55
34:A8:4E:FD:68:EE	-80
34:A8:4E:FD:68:EF	-79
54:78:1A:21:3B:C0	-80
54:78:1A:21:3B:C1	-81
54:78:1A:73:A2:60	-49
54:78:1A:73:A2:61	-49
54:78:1A:73:A2:6E	-55
54:78:1A:73:A2:6F	-55

Table 5.3 Position request at Test point 1

Position request consists of 11 Wi-Fi beacon MAC ids and corresponding RSS values. Data stored for Block 9 which is used in block identification is shown in table 5.4. The data shown in the table consist of Wi-Fi beacon MAC Id, propagation model for corresponding Wi-Fi beacon, minimum and maximum RSS values encountered for corresponding Wi-Fi beacon in Block 9. Block identification involves substituting RSS values from position request in corresponding propagation model along with distance from Wi-Fi beacon to centre of the block. If propagation models from correct block are used, the residual would be small.

MAC Id	Propagation model	Min RSS	Max RSS	Block Lat	Block Lon
34:A8:4E:FD:68:E0	y = -4.4049x - 60.178	-69	-28	55.92291	-3.17172
34:A8:4E:FD:68:E1	y = -3.4796x - 60.38	-70	-28	55.92291	-3.17172
34:A8:4E:FD:68:EE	y = -2.9643x - 79.945	-87	-28	55.92291	-3.17172
34:A8:4E:FD:68:EF	y = -4.5126x - 78.726	-87	-28	55.92291	-3.17172
54:78:1A:21:3B:C0	y = -20.591x - 57.123	-88	-40	55.92291	-3.17172
54:78:1A:21:3B:C1	y = -11.185x - 68.618	-86	-40	55.92291	-3.17172
54:78:1A:73:A2:60	y = -9.9899x - 44.117	-57	-24	55.92291	-3.17172
54:78:1A:73:A2:61	y = -10.08x - 45.787	-59	-24	55.92291	-3.17172
54:78:1A:73:A2:6E	y = -9.137x - 54.135	-67	-24	55.92291	-3.17172
54:78:1A:73:A2:6F	y = -8.342x - 54.425	-67	-24	55.92291	-3.17172

Table 5.4 Data stored for Block 9

The calculation of Euclidean for Block 9 with position request from test point 1 is summarized in table 5.5. The residuals for each Wi-Fi beacon are also shown in the table. The ‘Block distance’ for Block 9 is square root of squares of the residuals for each Wi-Fi beacon. The ‘block distance’ for block 9 from position request at test point 1 is 38.68.

MAC Id	Propagation model	Residual
00:23:5E:95:E4:E1	y+14.313x+67.509=0	-4.2060
34:A8:4E:FD:68:E0	y+4.4049x+60.178=0	11.1809
34:A8:4E:FD:68:E1	y+3.4796x+60.38=0	10.4029
34:A8:4E:FD:68:EE	y +2.9643x +79.945=0	4.9081
34:A8:4E:FD:68:EF	y +4.5126x +78.726=0	6.2974
54:78:1A:21:3B:C0	y +20.591x +57.123 =0	-6.6961
54:78:1A:21:3B:C1	y +11.185x +68.618=0	-3.3607
54:78:1A:73:A2:60	y +9.9899x +44.117=0	15.2534
54:78:1A:73:A2:61	y+10.08x +45.787=0	17.1070
54:78:1A:73:A2:6E	y+9.137x +54.135=0	17.9777
54:78:1A:73:A2:6F	y +8.342x + 54.425 =0	16.6476
Euclidean distance (Block distance)		38.6842

Table 5.5 Calculation of 'block distance' from test point 1

This process is repeated for all blocks in test area and corresponding block distances are calculated. Table 5.6 shows the 'block distances' from test point 1 to all blocks in test area and block with least block distance is highlighted in green. Block with least block distance is the location of test point. In the current example, Test point 1 returns least block distance for block 9.

Block	Block Distance
B1	207.4496
B2	173.2422
B3	107.3235
B4	257.6673
B5	260.4803
B6	267.1142
B7	310.2946
B8	119.8847
B9	38.6842
B10	94.634
B11	182.668

Table 5.6 Block distances for Test point 1

The block identification process described above is repeated for 11 blocks. Figure 5.30 shows test point 1's block identification information for other blocks. The percentage of matches found and calculated block distances from each block is plotted. Least block distance and highest Wi-Fi beacon match are highlighted in figure 5.30.

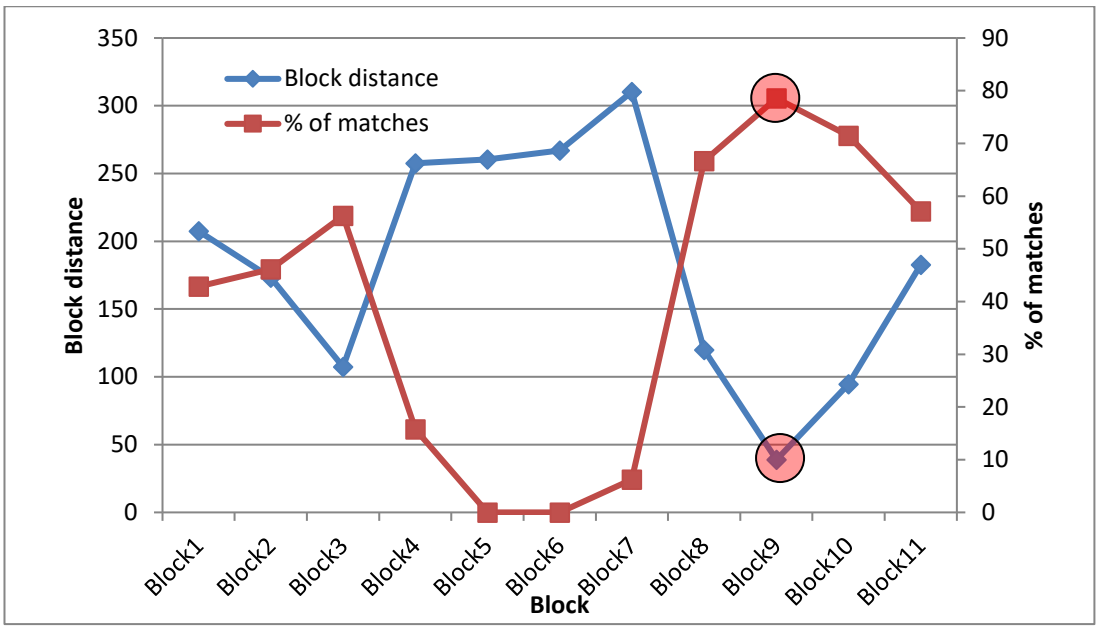


Figure 5.30 Test point 1 block identification data

5.10.1.1 Block identification errors

The block identification process detailed in the previous section has resulted in wrong identifications at four test points in the current test area. Consider test point 8 and 10, where block identification was incorrect. Table 5.7 shows block distances calculated for test points 8 and 10.

TP8	Block distance	Number of Beacons	Number of matches	% of matches
B1	47.13946	10	8	80
B2	77.6455	10	8	80
B3	83.57575	16	8	50
B4	162.9954	19	2	10.52632
B5	208.0865	16	0	0
B6	216.3331	18	0	0
B7	267.3948	16	0	0
B8	185.4999	16	7	43.75
B9	190.363	15	8	53.33333
B10	90.04646	12	10	83.33333
B11	47.88065	10	10	100
TP10	Block distance	Number of Beacons	Number of matches	% of matches
B1	147.394	14	0	0
B2	105.9481	14	0	0
B3	106.2169	18	2	11.11111
B4	79.31753	14	3	21.42857
B5	9.171431	6	6	100
B6	15.24892	8	6	75
B7	42.11778	8	4	50
B8	111.6915	19	0	0
B9	226.2189	19	0	0
B10	230.9221	18	0	0
B11	214.3012	16	0	0

Table 5.7 Test point 8 and 10 block distances

Test point 8 has shortest block distance to block 1 which is approximately 47.1 and next closest block is block 11 with 47.8. The actual block in which test point 8 was collected was block 11. It can be noticed that block 1 has 80% of matching beacons compared to 100% from block 11. Due to two missing Wi-Fi beacons in the position request scan, resulting distance to block 1 was less than distance to block 11. Test point 10 also has incorrect block identification due to missing beacons in position request scan. The correct block (B6) has only 75% matches compared to 100% matches in block 5. It can be noticed in figure 5.2 that incorrect blocks are adjacent to correct blocks.

Errors similar to test point 8 cases can be eliminated by using additional information such as number of beacons in position request, number of beacons detected in the block during calibration phase and percentage of matches. The errors

in cases similar to test point 10 can be eliminated to some extent by refining the penalties for missing beacons, however it would be difficult to eliminate such errors completely.

5.10.2 Lateration

The current lateration process was implemented in Matlab using non-linear least squares function. Non-linear least square function (lsqnonlin [89]) solves equation of the form

$$Min \| f(p) \|^2 = Min(f_1(p)^2 + f_2(p)^2 + ... + f_n(p)^2) \quad \ldots\ldots (5.9)$$

Where $| f(p) \|^2$ is the sum of squares of residuals of functions (f_1, f_2, …) and $f(p)$ is provided as input in a vector format which consists of residuals from individual functions represented in equation (5.10).

$$F = f(p) = \begin{bmatrix} f_1(p) \\ f_2(p) \\ ... \\ f_n(p) \end{bmatrix} \quad \ldots\ldots (5.10)$$

In Matlab, when using function 'lsqnonlin' the syntax is

$$p = lsqnonlin(fun, p_0, lb, ub, options) \quad \ldots\ldots (5.11)$$

'fun' in equation (5.11) is a matlab function which takes solution 'p' as input and returns 'F', which is residuals vector ' $f(p)$ ' which is evaluated at solution 'p'. 'p_0' in equation (5.11) is initial guess which needs to be input to the function. The better an initial guess is, the more chances of obtaining a better solution instead of being stuck at local minima. 'lb' is lower bound and 'ub' is upper bound of the estimated solution. 'options' in equation (5.11) is a parameter which is used to set options such as algorithm used and optimization parameters for algorithm used.

The residual functions in the current implementation are equations of circles, and have two variables (x, y). The solution (p and p_0) represented in equations (5.10) and (5.11) is intersection of circles represented as residual functions. The residual functions for current least squares problem were represented earlier in the equation (2.9) for a three-dimensional system. However, since current implementation is two-

dimensional local coordinate system, the residual function can be rewritten as follows

$$f_i(x,y) = \sum_{i=1}^{n} (\sqrt{(x-x_i)^2 + (y-y_i)^2} - r_i)^2 \ldots\ldots (5.12)$$

Where x_i and y_i are the coordinates of centre of circle 'i' with radius r_i.
In the current implementation of function 'lsqnonlin' in Matlab, 'options' were set to use 'trusted region reflective algorithm' [90] which is based on Newton method. The other settings in 'options' is to indicate if Jacobain of residual functions is computed in function 'fun' which one of the input parameters described in equation (5.11). A Jacobian matrix is first degree partial derivatives of variables in residual function which can be represented as follows

$$J = \begin{bmatrix} \frac{\partial f_1(x,y)}{\partial x} & \frac{\partial f_1(x,y)}{\partial y} \\ \frac{\partial f_2(x,y)}{\partial x} & \frac{\partial f_2(x,y)}{\partial y} \\ \ldots & \ldots \\ \frac{\partial f_n(x,y)}{\partial x} & \frac{\partial f_n(x,y)}{\partial y} \end{bmatrix} \quad \ldots\ldots \quad (5.12)$$

Based on the current implementation, possible reasons for errors in lateration algorithm are analysed in the following section.

5.10.2.1 Lateration errors

One of the main reasons for large errors in lateration results is errors in distance estimation to Wi-Fi beacon. This happens in two scenarios, first, incorrect block identification (discussed in section 5.10) which result in using incorrect propagation models, there by introducing errors in distance estimation. Second, the RSS value measured in position request is not described efficiently by the propagation model stored for the beacon for that block, leading to errors in distance estimation. Finally, Lateration errors also arise due to the distribution of Wi-Fi beacons. The reasons mentioned above are analysed in the following sections.

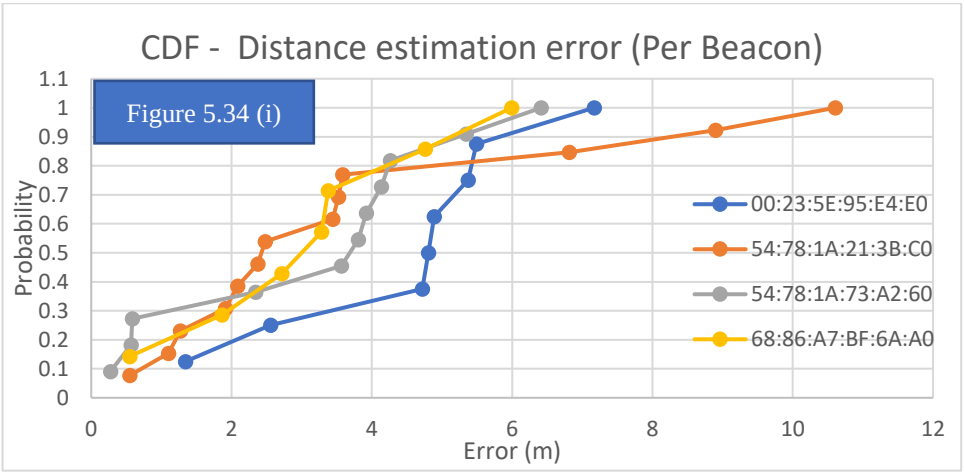
5.10.2.2 Distance estimation errors

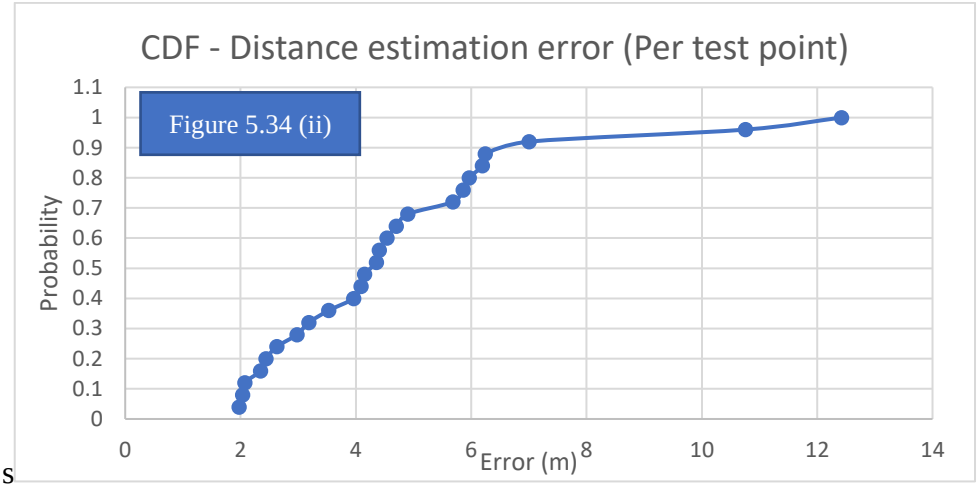
There were four test points for which block identification was incorrect (Test point 8, 10, 13 and 17). For some of these test points even using propagation models from wrong block have not affected lateration accuracy significantly. This proves the robustness of non-linear least squares algorithm. This also proves the effectiveness of

proposed coordinate transformation algorithm between latitude-longitude coordinates and local two-dimensional coordinates. As long as distance estimation errors are not very large, mobile device location estimation is accurate. Test points 8 and 10 have accuracies 0.7m and 1.3m respectively even though block identification was incorrect.

To discuss the second reason for inaccurate distance estimation, consider test point 5. In case of test point 5, the block was correctly identified to be Block 10, but accuracy of estimated location was 15m which is well above the average accuracy. The RSS value in position request scan for Wi-Fi beacon '54:78:1A:73:A2:60' is '-74dB', but during RSS values during calibration phase were between -61dB and -71dB. The estimated distance for Wi-Fi beacon '54:78:1A:73:A2:60' with propagation model for block 10 (y = -13.064x - 52.265) and RSS value from position request scan (-74dB) was 54m whereas the actual distance was 14m. These errors can be eliminated by checking the minimum and maximum values used in realizing a propagation model with in a block and to exclude such Wi-Fi beacons in lateration process.

The CDF of distance estimation errors for some of the Wi-Fi beacons is plotted in figure.5.34(i). 90^{th} percentile distance estimation error for most of the beacons is between 5m to 8m. Also, the CDF of average distance estimation error at all the test points is plotted in figure 5.34(ii). The 90^{th} percentile of error in distance estimation at all the test points with topology-based blocks is 7m. The block sizes varied from approximately 10m × 2.5m to 12.5m×13m. the 90^{th} percentile accuracy with topological blocks is about 8m compared to 10m with regular blocks (5m size). This shows that topology-based blocks perform slightly better compared to regular sized blocks.





s

Figure 5.34 CDF of Distance Estimation error

A plot of log-distance vs RSS plot for "34:A8:4E:FD:68:E0" is shown in figure 5.35. Data collected in each block is identified with different colour.

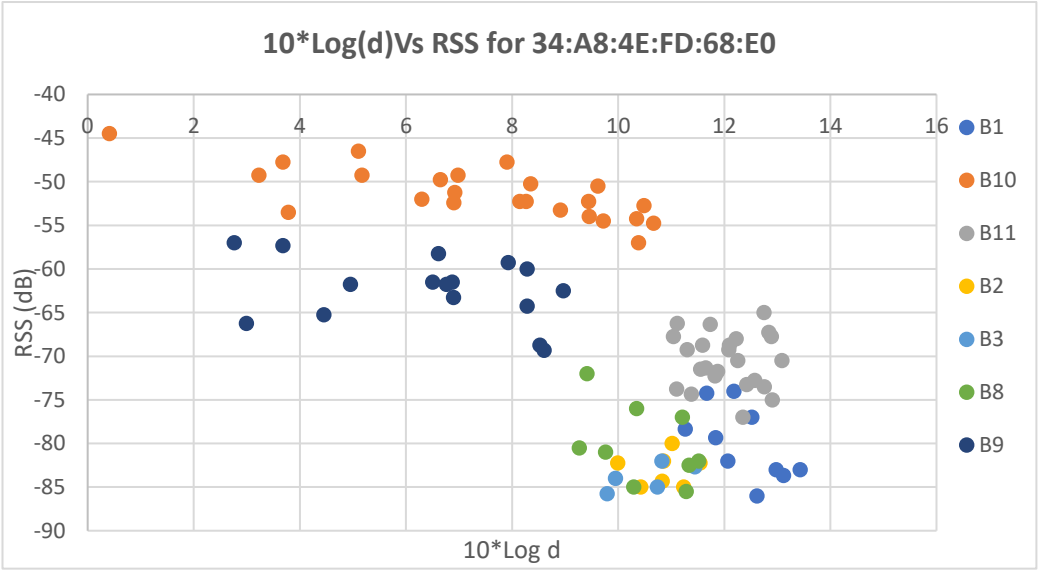


Figure 5.35 plot of 10*log(d) Vs RSS for Wi-Fi beacon "34:A8:4E:FD:68:E0"

There are three distinct patters in the plot. B10 (orange) and B9 (Blue) form separate clusters and it can be noticed in figure 5.21 that these two are adjacent class room with the beacon in B10. The beacon was installed on the wall between B9 and B 10. The signal manages to penetrate the wall and hence we see a reasonably strong RSS in B9. But is weaker in B11(Grey) as there is open space of the class room and a wall to reach B11. This shows that topology-based block sizes help in generating better propagation models. However, B11 is part of the third cluster, which would be difficult to pick automatically using the clustering algorithms evaluated in chapter 4.

5.10.2.3 Errors due to Wi-Fi beacon distribution

Some test points had large lateration errors due to Wi-Fi beacon distribution. This can be explained with test point 21 as an example. Figure 5.31 shows distribution Wi-Fi beacons used for lateration (P1, P2, P3 in green), actual location of mobile device (‘A’ in Red) and estimated location (‘E’ in Blue). The error in actual (‘A’) and estimated (‘E’) positions was approximately 15m. It can be noticed from figure 5.31, that all the Wi-Fi beacons were approximately to the south-east of actual location. Test point 21 indicates that Wi-Fi beacon distribution in the test area also has a significant impact over accuracy of estimated positions. A few cases similar to test point 21 were invested to determine influence of Wi-Fi beacon distribution over lateration accuracy.

Figure 5.31 Test point 21 Wi-Fi beacon, Actual and Estimated locations

To analyse the effect of Wi-Fi distribution on lateration accuracy, test scenarios with coverage on one side and at least two sides of actual location were investigated. The test data was collected at Hudson Beare building at University of Edinburgh. Figure 5.32 shows a scenario where the actual location is covered on two sides. In this scenario, lateration resulted in 2.5m accuracy. Points P1, P2, P3 and P4 in figure 5.32 represent Wi-Fi beacons; ‘A’ is the actual location and ‘E’ is estimated location.

Figure 5.32 Test case with coverage on two sides

The second scenario considered is that of coverage on one side of actual location. Figure 5.33 shows points representing Wi-Fi beacons (P1, P2, P3 and P4), actual location (‘A’) and estimated location (‘E’). Lateration accuracy in this scenario was 40m.

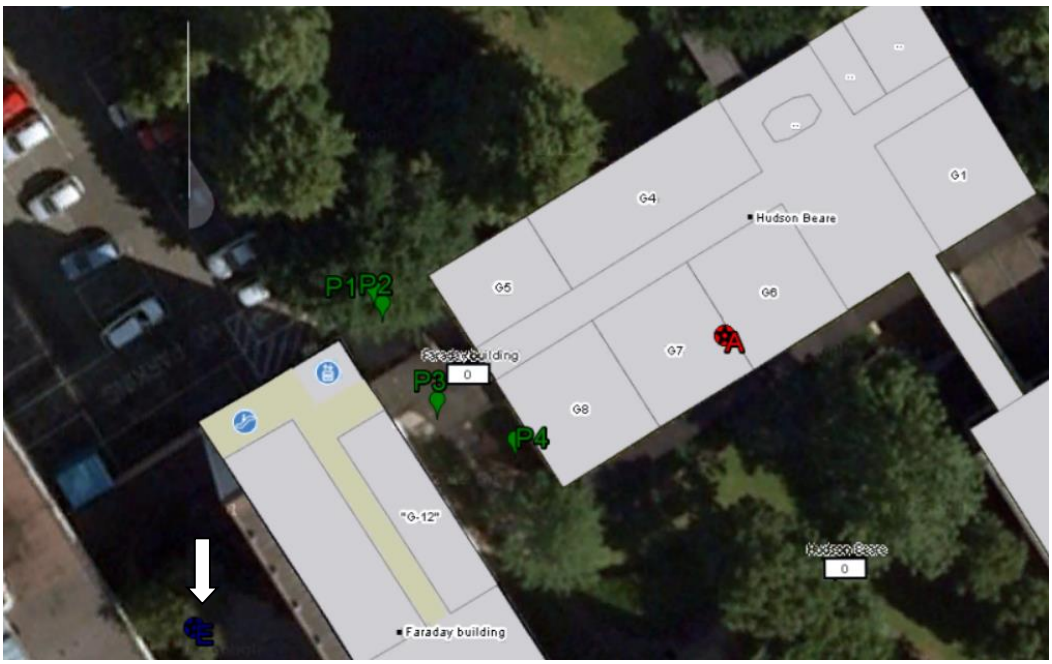


Figure 5.33 Test case with coverage on one side

5.11 Conclusion

Lateration:

Non-linear least squares algorithm for lateration is 20-30% better in terms of accuracy when compared to linearization algorithm. To achieve a good accuracy, non-linear least squares algorithm should be used, however, this algorithm adds computational intensity as it is iterative. Linearization algorithm does not suit systems which involve three dimensional coordinates. Linearization algorithm has an error of over 300m when estimating the height. Hence, non-linear least squares algorithm suits the systems involving three dimensional coordinates.

The second implementation which uses transformation algorithm to converts the latitude-longitude coordinates to local two-dimensional coordinates has been effective. The performance of lateration in terms of accuracy with the two-dimensional coordinates is almost same as three dimensional coordinates. However, the number of iterations was reduced by 57%, compared to three dimensional coordinates. Overall, the proposed transformation algorithm manages to deliver the same accuracy as three-dimensional coordinates but provides a better performance considering the reduction in number of computations required.

Lateration errors are due to two main reasons, Distance estimation errors and Wi-Fi beacon distribution. The distance estimation errors can be minimised by

checking if the propagation model can provide accurate results. If RSS value for a Wi-Fi beacon in position request is not within the range of values used for propagation model realization, it is likely that estimated distance would be inaccurate. However, the second reason, Wi-Fi beacon distribution is hard to overcome. The distribution is dependent on infrastructure installer and requirement of Wi-Fi internet coverage. In a scalable system, where the positioning system provider does not have any control over distribution of Wi-Fi beacons, these errors are even harder to overcome.

The 90th percentile accuracy using proposed positioning system was 11m, whereas the accuracy using nearest neighbour fingerprint system was 5m. The 11m accuracy 90% of the time is capable of supporting the use cases current work is targeting.

Database Size and Fingerprint algorithm for block identification:

The block identification in second implementation, does not rely on calibration scan collected at centre of the block. This reduces the amount of data needed to be stored. For the current test area, the reduction in data storage is up to 23 times compared to a fingerprint system. Bock sizes for the current implementation were based on the floor plan (topological blocks) instead of regular block sizes (such as 5m in previous implementation). In the second implementation, even though the test area covers almost 4 times the area covered in the first implementation, the number of blocks did not increase in the same proportion. The total number of blocks in the first implementation were 20 and in the second implementation total number of blocks is only 11. Also, the data that needs to be stored increased from 6.8KB in first implementation to 10.01KB in second implementation. The increase in the data size is due to the additional min RSS and max RSS values being stored and the number of beacons being considered in the second implementation covering a wider has also increased.

Distance estimation between the beacon and mobile device using propagation models specific to the topological blocks is between 5m to 6m. While the 90th percentile distance estimation error in the initial implementation with 5m regular blocks was 10m. This shows that using larger topological block sizes does not deteriorated the distance estimation.

Block identification using the ‘block distance’ was 84.6% successful which was 80% in the previous implementation. This proves that block identification using only propagation models for individual blocks is effective.

The block identification errors can be reduced by including the percentage of Wi-Fi beacon matches between position request and stored block data in block distance calculations. Along with refining the penalties added to the block distance for missing Wi-Fi beacons can help improve the block identification accuracy.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

The main aim of current work was to develop a scalable indoor positioning system which would require minimum database, minimizes computational intensity and an accuracy to support some of the use cases discussed in chapter 1. Positioning techniques that are widely implemented for indoor environments are Lateration and fingerprinting. Each system has its merits and drawbacks. The current work discussed drawbacks of each technique and a new hybrid positioning system was implemented.

Experimental system implementation presented in chapter 3 shows that instead of relying on a single propagation model to estimate the distances to Wi-Fi beacons, multiple models for smaller areas are more accurate. Data was collected at Sanderson building in test area which was divided into blocks and then processed to build propagation models. Distance estimations using block specific propagation models and a general propagation model for test area were compared. The 90^{th} percentile accuracy with block specific models was 4.3m compared to 17m using a single propagation model. The accuracy results confirm that using a propagation model specific to a block yields better distance estimations. System implementation in chapter 3 also compares the size of database for a fingerprint system and proposed hybrid system, which shows a reduction of up to eight times with the hybrid system. However, the system proposed in chapter 3 does not have a robust lateration

algorithm and also the coordinates used in the system were local three-dimensional coordinates which does not let the system be scalable.

Improvements to initially proposed system to further reduce the data stored and to make the system scalable were proposed in chapter 5. The improvements involved using latitude-longitude coordinates to make the system scalable and also implements a lateration algorithm which reduces the computational intensity. Lateration algorithm involves transforming latitude-longitude coordinates to Cartesian coordinates. Chapter 5 describes a transformation algorithm which converts latitude -longitude coordinates to two dimensional Cartesian coordinates and back to latitude-longitude coordinates. The improvements were implemented at Sanderson building. The improved system implementation reduced the data to be stored by about 20 times compared to a fingerprinting system. The number of comparisons required for block identification with the reduced database in the new hybrid system is about 15 times less than that would have been needed for a fingerprinting system. Also, lateration with two dimensional coordinates requires 57% less iterations compared to three dimensional coordinates. The 90^{th} percentile positioning accuracy of the final is 11m, which is acceptable for the use cases targeted.

The current accuracy of the hybrid system can support location-based use cases such as search, advertising and authentication. The reduction in the database size and computational requirements mean lower latency which helps support large number of users without needing large resources. A lower latency means, when a user starts a search on mobile device, its location is calculated with minimum latency and returns location aware results almost instantaneously. A smaller database can also let the positioning system be run on mobile phone by downloading the data for nearby area without having to connect to network for position estimation.

The current system uses the exiting Wi-Fi infrastructure and eliminating the additional costs. However, to support positioning in areas with poor or no Wi-Fi coverage, the BLE beacons presented in chapter 7 can be used. Beacon installation can improve coverage where needed with lower costs compared to installing Wi-Fi beacons.

One of the other aims of current work was also to automate the Wi-Fi beacon location information data collection. BLE beacon can help automate data collection process.

6.2 Future Work

The current work describes and implements a scalable hybrid positioning system which addresses some of the drawbacks of fingerprinting and lateration systems. However, the current system is still dependent on calibration phase during which data is collected at calibration points. One of the major drawbacks of current system is that any changes to Wi-Fi beacons would require recalibration of test area. Some of the scenarios where a complete recalibration would be required are relocation or replacement of Wi-Fi beacons or removal of some of the Wi-Fi beacons in the test area. Calibration of a test area is tedious and laborious process which takes time and resources. The future work should focus on making the calibration process simpler and minimising time taken in data collection.

The current system accuracy can be improved by optimising the fingerprint algorithm used in block identification. In the current implementation, 'block distance' is evaluated for all the blocks. Instead, the computation can be limited to the blocks which have at least 70 percent matching Wi-Fi beacons. Also, sorting algorithms can be used to speed up block distance calculation.

Lateration process in the current system was implemented in Matlab and uses non-linear least squares function which implements 'trusted region reflective algorithm' which is based on Newton method. However, Matlab allows using Levenberg-Marquardt algorithm. As part of future work, the performance of lateration with Levenberg-Marquardt algorithm can be investigated.

The calibration process in current implementation is time consuming and laborious. An individual has to stand at predetermined calibration point for up to a minute and repeat the process at all the calibration points. PDR (Pedestrian Dead Reckoning) can be used to automate the current calibration process. A user can input an initial location by touching on a map and then walk around in the block. PDR system would estimate the location and scan of Wi-Fi beacons can be tagged to the location. Multiple walks can be performed in each block and the raw data can be stored to be processed as described in earlier chapters.

An analysis of impact of block size on distance estimation, where the block sizes were varied from 5m to 20m and a propagation model was fitted for each block size. These models were then used to estimate the distances to the beacons from test points. For most of the beacons smallest sized returned the best distance estimations. Some of the propagation models were not good fit as there were environmental within a block. Automatic block size generation using clustering algorithms were evaluated, which indicate the block size can be automatically identified based on the calibration data. In current work only, K-means and DBSCAN clustering were evaluated. As part of future work, Other clustering algorithms can be evaluated.

Bluetooth low energy beacons in the current implementation, serve the purpose of transmitting location information of Wi-Fi beacons. These Bluetooth low energy beacons can be modified to transmit a location which can be used as initial location for the PDR system. An application can read the location beacon installed in each block, which can be used as initial location for PDR during calibration phase. A similar system using PDR for data collection is presented in [120]. A system with all the improvements integrated into a single application would reduce the time required and complexity involved in calibration phase significantly and also improves accuracy.

Bibliography

[1] Want, R., Hopper, A., Falcao, V. and Gibbons, J., 1992. The active badge location system. *ACM Transactions on Information Systems (TOIS), 10*(1), pp.91-102.
[2] Ni, L.M., Liu, Y., Lau, Y.C. and Patil, A.P., 2004. LANDMARC: indoor location sensing using active RFID. *Wireless networks, 10*(6), pp.701-710.
[3] Priyantha, N.B., Chakraborty, A. and Balakrishnan, H., 2000, August. The cricket location-support system. In *Proceedings of the 6th annual international conference on Mobile computing and networking* (pp. 32-43). ACM.
[4] Rabinowitz, M. and Spilker Jr, J.J., 2005. A new positioning system using television synchronization signals. *Broadcasting, IEEE Transactions on, 51*(1), pp.51-61.
[5] Papliatseyeu, A., Kotilainen, N., Mayora, O. and Osmani, V., 2009. FINDR: Low-cost indoor positioning using FM radio. In *MobileWireless Middleware, Operating Systems, and Applications* (pp. 15-26). Springer Berlin Heidelberg.
[6] Congshan, Q., Hualong, X. and Ying, T., 2007, May. Navigation and Positioning System Based on GPS & CDMA. In *Industrial Electronics and Applications, 2007. ICIEA 2007. 2nd IEEE Conference on* (pp. 2843-2846). IEEE.
[7] Bahl, P. and Padmanabhan, V.N., 2000. RADAR: An in-building RF-based user location and tracking system. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE* (Vol. 2, pp. 775-784). IEEE.
[8] Teuber, A. and Eissfeller, B., 2006, March. WLAN indoor positioning based on Euclidean distances and fuzzy logic. In *Proceedings of the 3rd Workshop on Positioning, Navigation and Communication* (pp. 159-168).
[9] Shuo, S., Hao, S. and Yang, S., 2010, December. Design of an experimental indoor position system based on RSSI. In *Information Science and Engineering (ICISE), 2010 2nd International Conference on* (pp. 1989-1992). IEEE.
[10] Gansemer, S., Großmann, U. and Hakobyan, S., 2010, September. Rssi-based euclidean distance algorithm for indoor positioning adapted for the use in dynamically changing wlan environments and multi-level buildings. In *Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on* (pp. 1-6). IEEE.
[11] Liu, H., Darabi, H., Banerjee, P. and Liu, J., 2007. Survey of wireless indoor positioning techniques and systems. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 37*(6), pp.1067-1080.
[12] Gu, Y., Lo, A. and Niemegeers, I., 2009. A survey of indoor positioning systems for wireless personal networks. *Communications Surveys & Tutorials, IEEE, 11*(1), pp.13-32.

[13] Hightower, J. and Borriello, G., 2001. Location systems for ubiquitous computing. *Computer*, (8), pp.57-66.
[14] Campos, R.S. and Lovisolo, L., 2015. *RF Positioning: Fundamentals, Applications, and Tools*. Artech House.
[15] Cheung, K.C., Intille, S.S. and Larson, K., 2006, September. An inexpensive bluetooth-based indoor positioning hack. In *Proceedings of UbiComp* (Vol. 6).
[16] Harter, A. and Hopper, A., 1994. A distributed location system for the active office. *Network, IEEE*, *8*(1), pp.62-70.
[17] Van Veen, B.D. and Buckley, K.M., 1988. Beamforming: A versatile approach to spatial filtering. *IEEE assp magazine*, *5*(2), pp.4-24.
[18] Stoica, P. and Moses, R.L., 1997. *Introduction to spectral analysis* (Vol. 1, pp. 3-4). Upper Saddle River: Prentice hall.
[19] Küpper, A., 2005. *Location-based services: fundamentals and operation*. John Wiley & Sons.
[20] Kontkanen, P., Myllymäki, P., Roos, T., Tirri, H., Valtonen, K. and Wettig, H., 2004, September. Topics in probabilistic location estimation in wireless networks. In *PIMRC* (pp. 1052-1056).
[21] Bahl, P., Padmanabhan, V.N. and Balachandran, A., 2000. *Enhancements to the RADAR user location and tracking system*. technical report, Microsoft Research.
[22] Prasithsangaree, P., Krishnamurthy, P. and Chrysanthis, P.K., 2002, September. On indoor position location with wireless LANs. In *Personal, Indoor and Mobile Radio Communications, 2002. The 13th IEEE International Symposium on* (Vol. 2, pp. 720-724). IEEE.
[23] Brunato, M. and Kiss Kallo, C., 2002. Transparent location fingerprinting for wireless services.
[24] Castro, P., Chiu, P., Kremenek, T. and Muntz, R., 2001, January. A probabilistic room location service for wireless networked environments. In *Ubicomp 2001: Ubiquitous Computing* (pp. 18-34). Springer Berlin Heidelberg.
[25] Roos, T., Myllymäki, P., Tirri, H., Misikangas, P. and Sievänen, J., 2002. A probabilistic approach to WLAN user location estimation. *International Journal of Wireless Information Networks*, *9*(3), pp.155-164.
[26] Honkavirta, V., Perälä, T., Ali-Löytty, S. and Piché, R., 2009, March. A comparative survey of WLAN location fingerprinting methods. In *Positioning, Navigation and Communication, 2009. WPNC 2009. 6th Workshop on* (pp. 243-251). IEEE.
[27] Peterson, B.B., Kmiecik, C., Hartnett, R., Thompson, P.M., Mendoza, J. and Nguyen, H., 1998. Spread spectrum indoor geolocation. *Navigation*, *45*(2), pp.97-102.
[28] Fontana, R.J. and Gunderson, S.J., 2002, May. Ultra-wideband precision asset location system. In *Ultra Wideband Systems and Technologies, 2002. Digest of Papers. 2002 IEEE Conference on* (pp. 147-150). IEEE.
[29] Li, X., Pahlavan, K., Latva-aho, M. and Ylianttila, M., 2000. Comparison of indoor geolocation methods in DSSS and OFDM wireless LAN systems. In *Vehicular Technology Conference, 2000. IEEE-VTS Fall VTC 2000. 52nd* (Vol. 6, pp. 3015-3020). IEEE.
[30] Günther, A. and Hoene, C., 2005. Measuring round trip times to determine the distance between WLAN nodes. In *NETWORKING 2005. Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks;*

Mobile and Wireless Communications Systems (pp. 768-779). Springer Berlin Heidelberg.
[31] Pahlavan, K., 2011. *Principles of wireless networks: A unified approach*. John Wiley & Sons, Inc..
[32] Castro, P., Chiu, P., Kremenek, T. and Muntz, R., 2001, January. A probabilistic room location service for wireless networked environments. In *Ubicomp 2001: Ubiquitous Computing* (pp. 18-34). Springer Berlin Heidelberg.
[33] Bardwell, J., 2004. You Believe You Understand What You Think I Said…. *The truth about, 802.11 signal and noise metrics*. Available online at: http://www.n-cg.net/ncgpdf/WiFi_SignalValues.pdf.
[34] Hightower, J., Want, R. and Borriello, G., 2000. SpotON: An indoor 3D location sensing technology based on RF signal strength. *UW CSE 00-02-02, University of Washington, Department of Computer Science and Engineering, Seattle, WA, 1*.
[35] Isaacson, E. and Keller, H.B., 2012. *Analysis of numerical methods*. Courier Corporation.
[36] Noble, B. and Daniel, J.W., 1988. *Applied linear algebra* (Vol. 3). New Jersey: Prentice-Hall.
[37] Bancroft, S., 1985. An algebraic solution of the GPS equations. *Aerospace and Electronic Systems, IEEE Transactions on*, (1), pp.56-59.
[38] Gander, W., 1980. Algorithms for the QR decomposition. *Res. Rep*, *80*(02), pp.1251-1268.
[39] Golub, G.H. and Reinsch, C., 1970. Singular value decomposition and least squares solutions. *Numerische mathematik*, *14*(5), pp.403-420.
[40] Foy, W.H., 1976. Position-location solutions by Taylor-series estimation. *IEEE Transactions on Aerospace and Electronic Systems*, pp.187-194.
[41] Navidi, W., Murphy, W.S. and Hereman, W., 1998. Statistical methods in surveying by trilateration. *Computational statistics & data analysis*, *27*(2), pp.209-227.
[42] Murphy, W. and Hereman, W., 1995. Determination of a position in three dimensions using trilateration and approximate distances. *Department of Mathematical and Computer Sciences, Colorado School of Mines, Golden, Colorado, MCS-95*, *7*, p.19.
[43] Fang, B.T., 1990. Simple solutions for hyperbolic and related position fixes. *Aerospace and Electronic Systems, IEEE Transactions on*, *26*(5), pp.748-753.
[44] Pahlavan, K. and Kanaan, M., 2004. A comparison of Wireless geolocation algorithms in indoor environments. In *Wireless Communication and Networking Conference, WCNC IEEE* (Vol. 1, pp. 21-25).
[45] Madsen, K., Bruun, H. and Tingleff, O., 1999. Methods for non-linear least squares problems. Tech. rep., Informatics and Mathematical Modelling, Technical University of Denmark.
[46] Kaemarungsi, K. and Krishnamurthy, P., 2004, August. Properties of indoor received signal strength for WLAN location fingerprinting. In *Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004. The First Annual International Conference on* (pp. 14-23). IEEE.
[47] Kaemarungsi, K., 2006, January. Distribution of WLAN received signal strength indication for indoor location determination. In *Wireless Pervasive Computing, 2006 1st International Symposium on* (pp. 6-pp). IEEE.

[48] Ladd, A.M., Bekris, K.E., Rudys, A., Kavraki, L.E. and Wallach, D.S., 2005. Robotics-based location sensing using wireless ethernet. *Wireless Networks*, *11*(1-2), pp.189-204.
[49] Faruque, S., 2014. *Radio Frequency Propagation Made Easy*. Springer.
[50] Seybold, J.S., 2005. *Introduction to RF propagation*. John Wiley & Sons.
[51] Rappaport, T.S., 1996. *Wireless communications: principles and practice* (Vol. 2). New Jersey: prentice hall PTR.
[52] Sarkar, T.K., Ji, Z., Kim, K., Medouri, A. and Salazar-Palma, M., 2003. A survey of various propagation models for mobile communication. *Antennas and Propagation Magazine, IEEE*, *45*(3), pp.51-82.
[53] Sharma, P.K. and Singh, R.K., 2010. Comparative analysis of propagation path loss models with field measured data. *International Journal of Engineering Science and Technology*, *2*(6), pp.2008-2013.
[54] Hashemi, H., 1993. The indoor radio propagation channel. *Proceedings of the IEEE*, *81*(7), pp.943-968.
[55] T. Holt, I. C. Pahlavan, and J. F. Lee, 1992 October, Ray tracing algorithm for indoor radio propagation modeling. 3rd IEEE Int. Symp. on Personal, Indoor and Mobile Radio Communications, IEEE.
[56] McKown, J.W. and Hamilton, R.L., 1991. Ray tracing as a design tool for radio networks. *Network, IEEE*, *5*(6), pp.27-30.
[57] Seidel, S.Y. and Rappaport, T.S., 1992, December. A ray tracing technique to predict path loss and delay spread inside buildings. In *Global Telecommunications Conference, 1992. Conference Record., GLOBECOM'92. Communication for Global Users., IEEE* (pp. 649-653). IEEE.
[58] Valenzuela, R., 1993, May. A ray tracing approach to predicting indoor wireless transmission. In *Vehicular Technology Conference, 1993., 43rd IEEE* (pp. 214-218). IEEE.
[59] Hata, M., 1980. Empirical formula for propagation loss in land mobile radio services. *Vehicular Technology, IEEE Transactions on*, *29*(3), pp.317-325.
[60] Seidel, S.Y. and Rappaport, T.S., 1992. 914 MHz path loss prediction models for indoor wireless communications in multifloored buildings. *Antennas and Propagation, IEEE Transactions on*, *40*(2), pp.207-217.
[61] www.lx.it.pt. 2003. *Final report*. [ONLINE] Available at: http://www.lx.it.pt. [Accessed October 2015].
[62] www.itu.int. 2003. *ITU recommendations*. [ONLINE] Available at: https://www.itu.int/dms_pubrec/itu-r/rec/p/R-REC-P.1238-0-199705-S!!PDF-E.pdf. [Accessed October 2015].
[63] Devasirvatham, D.M., 1984. Time delay spread measurements of wideband radio signals within a building. *Electronics Letters*, *20*(23), pp.950-951.
[64] S. Saunders, 2000, “Antennas and Propagation for WirelessCommunication Systems”, Wiley.
[65] Lott, M. and Forkel, I., 2001. A multi-wall-and-floor model for indoor radio propagation. In *Vehicular Technology Conference, 2001. VTC 2001 Spring. IEEE VTS 53rd* (Vol. 1, pp. 464-468). IEEE.
[66] Milanovic, J., Rimac-Drlje, S. and Bejuk, K., 2007, December. Comparison of propagation models accuracy for WiMAX on 3.5 GHz. In *Electronics, Circuits and Systems, 2007. ICECS 2007. 14th IEEE International Conference on* (pp. 111-114). IEEE.

[67] Abhayawardhana, V.S., Wassell, I.J., Crosby, D., Sellars, M.P. and Brown, M.G., 2005, May. Comparison of empirical propagation path loss models for fixed wireless access systems. In *Vehicular Technology Conference, 2005. VTC 2005-Spring. 2005 IEEE 61st* (Vol. 1, pp. 73-77). IEEE.
[68] Robusto, C.C., 1957. The cosine-haversine formula. *American Mathematical Monthly*, pp.38-40.
[69] Kutner, M.H., 1996. *Applied linear statistical models* (Vol. 4). Chicago: Irwin.
[70] Seber, G.A. and Lee, A.J., 2012. *Linear regression analysis* (Vol. 936). John Wiley & Sons.
[71] Erceg, V., Greenstein, L.J., Tjandra, S.Y., Parkoff, S.R., Gupta, A., Kulic, B., Julius, A. and Bianchi, R., 1999. An empirically based path loss model for wireless channels in suburban environments. *Selected Areas in Communications, IEEE Journal on*, *17*(7), pp.1205-1211.
[72] Deza, M.M. and Deza, E., 2009. *Encyclopedia of distances* (pp. 1-583). Springer Berlin Heidelberg.
[73] http://www.inssider.com/downloads/ [Accessed October 2015].
[74] Faruque, S., 2014. *Radio Frequency Propagation Made Easy*. Springer.
[75] Aragon-Zavala, A., 2008. *Antennas and propagation for wireless communication systems*. John Wiley & Sons.
[76] Kjærgaard, M.B., 2011. Indoor location fingerprinting with heterogeneous clients. *Pervasive and Mobile Computing*, *7*(1), pp.31-43.
[77] Kennedy, M. and Koop, S., 1994. Understanding map projections. *GIS by ESRI, Environmental System Research Institute*.
[78] Featherstone, W.E. and Claessens, S.J., 2008. Closed-form transformation between geodetic and ellipsoidal coordinates. *Studia geophysica et geodaetica*, *52*(1), pp.1-18.
[79] Johns, R.K.C., 1959. The Figure of the Earth. *Journal of the Royal Astronomical Society of Canada*, *53*, p.257.
[80] www.ordnancesurvey.co.uk, Guide to coordinate systems in Great Britain, [ONLINE], available at: https://www.ordnancesurvey.co.uk/docs/support/guide-coordinate-systems-great-britain.pdf , [Accessed October 2015].
[81] Ligas, M. and Banasik, P., 2011. Conversion between Cartesian and geodetic coordinates on a rotational ellipsoid by solving a system of nonlinear equations. *Geodesy and Cartography*, *60*(2), pp.145-159.
[82] Maling, D.H., 1968. Coordinate Systems and Map Projections for GIS.
[83] Kennedy, M. and Koop, S., 1994. Understanding map projections. *GIS by ESRI, Environmental System Research Institute*.
[84] Admiralty, G.B., 1915. Admiralty manual of navigation, 1914.
[85] Bullock, R., 2007. Great circle distances and bearings between two locations. *MDT*, *June*, *5*.
[86] Robusto, C.C., 1957. The cosine-haversine formula. *American Mathematical Monthly*, pp.38-40.
[87] Williams, E., 2014. Aviation Formulary V1. 46., [ONLINE], available at: http://williams.best.vwh.net/avform.htm , [Accessed October 2015].
[88] Veness, C., 2010. Calculate distance, bearing and more between Latitude/Longitude points, [ONLINE], available at: http://www.movable-type.co.uk/scripts/latlong.html , [Accessed October 2015].

[89] Mathworks , 2010. lsqnonlin solve nonlinear, [ONLINE], available at: http://uk.mathworks.com/help/optim/ug/lsqnonlin.html, [Accessed October 2015].
[90] Coleman, T.F. and Li, Y., 1996. An interior trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on optimization*, *6*(2), pp.418-445.
[91] Huang, A.S. and Rudolph, L., 2007. *Bluetooth essentials for programmers*. Cambridge University Press.
[92] Heydon, R., 2012. *Bluetooth low energy: the developer's handbook*. Prentice Hall.
[93] Hunn, N., 2010. *Essentials of Short-Range Wireless*. Cambridge University Press.
[94] bluetooth.com, List of adopted specifications, [ONLINE], available at: https://www.bluetooth.com/specifications/adopted-specifications, [Accessed October 2015].
[95] Chakraborty, G., Naik, K., Chakraborty, D., Shiratori, N. and Wei, D., 2010. Analysis of the Bluetooth device discovery protocol. *Wireless Networks*, *16*(2), pp.421-436.
[96] Bray, J., Kammer, D., McNutt, G. and Senese, B., 2002. *Bluetooth application developer's guide: The short range interconnect solution*. Syngress publishing.
[97] Soh, W.S., 2007, September. A comprehensive study of bluetooth signal parameters for localization. In *Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007. IEEE 18th International Symposium on* (pp. 1-5). IEEE.
[98] Figueiras, J., Schwefel, H.P. and Kovacs, I., 2005, September. Accuracy and timing aspects of location information based on signal-strength measurements in Bluetooth. In *Personal, Indoor and Mobile Radio Communications, 2005. PIMRC 2005. IEEE 16th International Symposium on* (Vol. 4, pp. 2685-2690). IEEE.
[99] Cheung, K.C., Intille, S.S. and Larson, K., 2006, September. An inexpensive bluetooth-based indoor positioning hack. In *Proceedings of UbiComp* (Vol. 6).
[100] Hallberg, J., Nilsson, M. and Synnes, K., 2003, March. Positioning with bluetooth. In *Telecommunications, 2003. ICT 2003. 10th International Conference on* (Vol. 2, pp. 954-958). IEEE.
[101] Genco, A., 2005. Three step bluetooth positioning. In *Location-and Context-Awareness* (pp. 52-62). Springer Berlin Heidelberg.
[102] Rodriguez, M., Pece, J.P. and Escudero, C.J., 2005, May. In-building location using bluetooth. In *International Workshop on Wireless Ad-hoc Networks*.
[103] Aparicio, S., Pérez, J., Bernardos, A.M. and Casar, J.R., 2008, August. A fusion method based on Bluetooth and WLAN technologies for indoor location. In *Multisensor Fusion and Integration for Intelligent Systems, 2008. MFI 2008. IEEE International Conference on* (pp. 487-491). IEEE.
[104] ti.com, Bluetooth low energy chips overview, [ONLINE], available at: http://www.ti.com/lsds/ti/wireless_connectivity/bluetooth_bluetooth-ble/overview.page, [Accessed October 2015].
[105]]nordicsemi.com, Bluetooth low energy chips overview, [ONLINE], available at: https://www.nordicsemi.com/eng/Products/Bluetooth-Smart-Bluetooth-low-energy, [Accessed October 2015].
[106] android.com, Bluetooth low energy developer overview, [ONLINE], available at: http://developer.android.com/guide/topics/connectivity/bluetooth-le.html, [Accessed October 2015].

[107] apple.com, core Bluetooth documentation programming guide, [ONLINE], available at: https://developer.apple.com/library/ios/ /NetworkingInternetWeb/Conceptual/CoreBluetooth_concepts/AboutCoreBluetooth/ Introduction.html, [Accessed October 2015].
[108] apple.com, iBeacon for developers, [ONLINE], available at: https://developer.apple.com/ibeacon/, [Accessed October 2015].
[109] android.com, Beacons, [ONLINE], available at: https://developers.google.com/beacons/, [Accessed October 2015].
[110] estimate .com, Overview of BLE beacon products, [ONLINE], available at: http://estimote.com/#jump-to-products, [Accessed October 2015].
[111] kontakt.io, Overview of BLE beacon products, [ONLINE], available at: https://store.kontakt.io/, [Accessed October 2015].
[112] radiusnetworks.com, Overview of BLE beacon products, [ONLINE], available at: http://store.radiusnetworks.com/collections/all, [Accessed October 2015].
[113] ti.com, CC2541 mini development kit, [ONLINE], available at: http://www.ti.com/tool/cc2541dk-mini, [Accessed October 2015].
[114] ti.com, CC2540 and CC2541 Bluetooth low energy software developer's reference guide, [ONLINE], available at: http://www.ti.com/lit/ug/swru271g/swru271g.pdf, [Accessed October 2015].
[115] ti.com, Bluetooth low energy software stack, [ONLINE], available at: http://www.ti.com/tool/ble-stack, [Accessed October 2015].
[116] android.com, Android studio overview, [ONLINE], available at: http://developer.android.com/tools/studio/index.html, [Accessed October 2015].
[117] Zhang, M., Shen, W. and Zhu, J., 2016, May. WIFI and magnetic fingerprint positioning algorithm based on KDA-KNN. In *Control and Decision Conference (CCDC), 2016 Chinese* (pp. 5409-5415). IEEE.
[118] Li, D., Zhang, B., Yao, Z. and Li, C., 2014, December. A feature scaling based k-nearest neighbor algorithm for indoor positioning system. In *2014 IEEE Global Communications Conference* (pp. 436-441). IEEE.
[119] Wang, B., Zhou, S., Liu, W. and Mo, Y., 2015. Indoor localization based on curve fitting and location search using received signal strength. *IEEE Transactions on Industrial Electronics*, *62*(1), pp.572-582.
[120] Liu, H.H., Liao, C.W. and Lo, W.H., 2015, August. The fast collection of radio fingerprint for WiFi-based indoor positioning system. In *Heterogeneous Networking for Quality, Reliability, Security and Robustness (QSHINE), 2015 11th International Conference on* (pp. 427-432). IEEE.
[121] Mobile and tablet internet usage exceeds desktop for first time worldwide[online]http://gs.statcounter.com/press/mobile-and-tablet-internet-usage-exceeds-desktop-for-first-time-worldwide [accessed on 06/03/2018]
[122] Internet access – households and individuals: 2017 available [online] at https://www.ons.gov.uk/peoplepopulationandcommunity/householdcharacteristics/homeinternetandsocialmediausage/bulletins/internetaccesshouseholdsandindividuals/2017#quality-and-methodology [accessed on 06/03/2018]
[123] Klepeis, N.E., Nelson, W.C., Ott, W.R., Robinson, J.P., Tsang, A.M., Switzer, P., Behar, J.V., Hern, S.C. and Engelmann, W.H., 2001. The National Human Activity Pattern Survey (NHAPS): a resource for assessing exposure to environmental pollutants. *Journal of Exposure Science and Environmental Epidemiology*, *11*(3), p.231.

[124] Location-Targeted Mobile Ad Spend to Reach $29.5B in the U.S. in 2020[Online] Availabe at :http://www.biakelsey.com/location-targeted-mobile-ad-spend-reach-29-5b-u-s-2020/ [Accessed on 06/03/2018]
[125] 48% of Consumers Spend More After Personalized E-commerce Efforts, Survey Says[Online] Available at : https://www.targetmarketingmag.com/article/48-consumers-spend-more-after-personalized-e-commerce-efforts-survey/all/ [Accessed on 06/03/2018]
[126] Fraud the facts 2017- The definitive overview of payment industry fraud [Online]Available at : https://www.financialfraudaction.org.uk/fraudfacts17/assets/fraud_the_facts.pdf [Accessed on 06/03/2018]
[127] Mobile Payments Rise in Popularity, Reaching Tipping Point in Some Countries, Global Report by ACI Worldwide Finds [Online] Available at https://www.aciworldwide.com/news-and-events/press-releases/2017/september/mobile-payments-rise-in-popularity-reaching-tipping-point-in-some-countries [Accessed on 06/03/2018]
[128] Mautz, R., 2012. Indoor positioning technologies, https://www.research-collection.ethz.ch/bitstream/handle/20.500.11850/54888/eth-5659-01.pdf.
[129] Wirola, L., Laine, T.A. and Syrjärinne, J., 2010, September. Mass-market requirements for indoor positioning and indoor navigation. In *Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on* (pp. 1-7). IEEE
[130] Lin, T.N. and Lin, P.C., 2005, June. Performance comparison of indoor positioning techniques based on location fingerprinting in wireless networks. In *Wireless Networks, Communications and Mobile Computing, 2005 International Conference on* (Vol. 2, pp. 1569-1574). IEEE
[131] Mok, E. and Retscher, G., 2007. Location determination using WiFi fingerprinting versus WiFi trilateration. *Journal of Location Based Services*, *1*(2), pp.145-159.
[132] Bell, S., Jung, W.R. and Krishnakumar, V., 2010, November. WiFi-based enhanced positioning systems: accuracy through mapping, calibration, and classification. In *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness* (pp. 3-9). ACM
[133] Sciarrone, A., Fiandrino, C., Bisio, I., Lavagetto, F., Kliazovich, D. and Bouvry, P., 2016, October. Smart probabilistic fingerprinting for indoor localization over fog computing platforms. In *Cloud Networking (Cloudnet), 2016 5th IEEE International Conference on* (pp. 39-44). IEEE.
[134] Sakr, M. and El-Sheimy, N., 2017, September. Efficient Wi-Fi signal strength maps using sparse Gaussian process models. In *Indoor Positioning and Indoor Navigation (IPIN), 2017 International Conference on* (pp. 1-8). IEEE.
[135] Lymberopoulos, D., Liu, J., Yang, X., Choudhury, R.R., Handziski, V. and Sen, S., 2015, April. A realistic evaluation and comparison of indoor location technologies: Experiences and lessons learned. In *Proceedings of the 14th international conference on information processing in sensor networks* (pp. 178-189). ACM.
[136] Zhao, J., Gao, X., Wang, X., Li, C., Song, M. and Sun, Q., 2018. An Efficient Radio Map Updating Algorithm based on K-Means and Gaussian Process Regression. *The Journal of Navigation*, pp.1-14.

[137] Kjærgaard, M.B., Krarup, M.V., Stisen, A., Prentow, T.S., Blunck, H., Grønbæk, K. and Jensen, C.S., 2013, October. Indoor positioning using wi-fi–how well is the problem understood?. In *International Conference on Indoor Positioning and Indoor Navigation* (Vol. 28, p. 31st).
[138] Mathisen, A., Sørensen, S.K., Stisen, A., Blunck, H. and Grønbæk, K., 2016, October. A comparative analysis of Indoor WiFi Positioning at a large building complex. In *Indoor Positioning and Indoor Navigation (IPIN), 2016 International Conference on* (pp. 1-8). IEEE.
[139] Qi, H. and Moore, J.B., 2002. Direct Kalman filtering approach for GPS/INS integration. *IEEE Transactions on Aerospace and Electronic Systems*, *38*(2), pp.687-693.
[140] Drake, S.P., 2002. Converting GPS coordinates [phi, lambda, h] to navigation coordinates (ENU).
[141] Qi, H. and Moore, J.B., 2002. Direct Kalman filtering approach for GPS/INS integration. *IEEE Transactions on Aerospace and Electronic Systems*, *38*(2), pp.687-693.
[142] Knauth, S., 2017, September. Smartphone PDR positioning in large environments employing WiFi, particle filter, and backward optimization. In *Indoor Positioning and Indoor Navigation (IPIN), 2017 International Conference on* (pp. 1-6). IEEE.
[143] Zhuang, Y., Lan, H., Li, Y. and El-Sheimy, N., 2015. PDR/INS/WiFi integration based on handheld devices for indoor pedestrian navigation. *Micromachines*, *6*(6), pp.793-812.
[144] Lin, K., Kansal, A., Lymberopoulos, D. and Zhao, F., 2010, June. Energy-accuracy trade-off for continuous mobile device location. In *Proceedings of the 8th international conference on Mobile systems, applications, and services* (pp. 285-298). ACM.
[145] Zhuang, Z., Kim, K.H. and Singh, J.P., 2010, June. Improving energy efficiency of location sensing on smartphones. In *Proceedings of the 8th international conference on Mobile systems, applications, and services* (pp. 315-330). ACM.
[146] Pathak, A., Hu, Y.C. and Zhang, M., 2012, April. Where is the energy spent inside my app?: fine grained energy accounting on smartphones with eprof. In *Proceedings of the 7th ACM european conference on Computer Systems* (pp. 29-42). ACM.
[147] Bisio, I., Lavagetto, F., Marchese, M. and Sciarrone, A., 2016. Smart probabilistic fingerprinting for WiFi-based indoor positioning with mobile devices. *Pervasive and Mobile Computing*, *31*, pp.107-123.
[148] Juels, A., 2006. RFID security and privacy: A research survey. *IEEE journal on selected areas in communications*, *24*(2), pp.381-394.
[149] Hightower, J., Want, R. and Borriello, G., 2000. SpotON: An indoor 3D location sensing technology based on RF signal strength.
[150] Koyuncu, H. and Yang, S.H., 2010. A survey of indoor positioning and object locating systems. *IJCSNS International Journal of Computer Science and Network Security*, *10*(5), pp.121-128.
[151] Adame, T., Bel, A., Bellalta, B., Barcelo, J. and Oliver, M., 2014. IEEE 802.11 ah: the WiFi approach for M2M communications. *IEEE Wireless Communications*, *21*(6), pp.144-152.

[152] Shen, Y. and Win, M.Z., 2010. Fundamental limits of wideband localization—Part I: A general framework. *IEEE Transactions on Information Theory*, *56*(10), pp.4956-4980.
[153] Belloni, F., Ranki, V., Kainulainen, A. and Richter, A., 2009, March. Angle-based indoor positioning system for open indoor environments. In *Positioning, Navigation and Communication, 2009. WPNC 2009. 6th Workshop on* (pp. 261-265). IEEE.
[154] D. Bharadia, E. McMilin, and S. Katti, "Full duplex radios," inProc.ACM SIGCOMM, 2013, pp. 375–386.
[155] Day, B.P., Margetts, A.R., Bliss, D.W. and Schniter, P., 2012. Full-duplex bidirectional MIMO: Achievable rates under limited dynamic range. *IEEE Transactions on Signal Processing*, *60*(7), pp.3702-3713.
[156] Dawes, B. and Chin, K.W., 2011. A comparison of deterministic and probabilistic methods for indoor localization. *Journal of Systems and Software*, *84*(3), pp.442-451.
[157] Boonsriwai, S. and Apavatjrut, A., 2013, May. Indoor WIFI localization on mobile devices. In *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2013 10th International Conference on* (pp. 1-5). IEEE.
[158] Sánchez-Rodríguez, D., Hernández-Morera, P., Quinteiro, J.M. and Alonso-González, I., 2015. A low complexity system based on multiple weighted decision trees for indoor localization. *Sensors*, *15*(6), pp.14809-14829.
[159] Le Dortz, N., Gain, F. and Zetterberg, P., 2012, March. WiFi fingerprint indoor positioning system using probability distribution comparison. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on* (pp. 2301-2304). IEEE.
[160] Youssef, M.A., Agrawala, A., Shankar, A.U. and Noh, S.H., 2002. *A probabilistic clustering-based indoor location determination system*
[161] Li, Y., Williams, S., Moran, B. and Kealy, A., 2018. Quantized RSS Based Wi-Fi Indoor Localization with Room Level Accuracy. In *Proceedings of the IGNSS 2018 Conference* (pp. 7-9).
[162] Youssef, M. and Agrawala, A., 2005, June. The Horus WLAN location determination system. In *Proceedings of the 3rd international conference on Mobile systems, applications, and services* (pp. 205-218). ACM.
[163] Kjærgaard, M.B., Treu, G. and Linnhoff-Popien, C., 2007, May. Zone-based rss reporting for location fingerprinting. In *International Conference on Pervasive Computing* (pp. 316-333). Springer, Berlin, Heidelberg.
[164] Brouwers, N., Zuniga, M. and Langendoen, K., 2014, March. Incremental wi-fi scanning for energy-efficient localization. In *Pervasive Computing and Communications (PerCom), 2014 IEEE International Conference on* (pp. 156-162). IEEE.
[165] Bisio, I., Lavagetto, F., Marchese, M. and Sciarrone, A., 2016. Smart probabilistic fingerprinting for WiFi-based indoor positioning with mobile devices. *Pervasive and Mobile Computing*, *31*, pp.107-123.
[166] Bisio, I., Lavagetto, F., Sciarrone, A. and Yiu, S., 2017, May. A Smart 2 Gaussian process approach for indoor localization with RSSI fingerprints. In *Communications (ICC), 2017 IEEE International Conference on* (pp. 1-6). IEEE.

[167] Llombart, M., Ciurana, M. and Barcelo-Arroyo, F., 2008, March. On the scalability of a novel WLAN positioning system based on time of arrival measurements. In *Positioning, Navigation and Communication, 2008. WPNC 2008. 5th Workshop on* (pp. 15-21). IEEE.
[168] Au, E., 2016. The latest progress on ieee 802.11 mc and ieee 802.11 ai [standards]. *IEEE Vehicular Technology Magazine*, *11*(3), pp.19-21.
[169] Kotaru, M. and Katti, S., 2017, July. Position Tracking for Virtual Reality Using Commodity WiFi. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
[170] Zhao, J., Gao, X., Wang, X., Li, C., Song, M. and Sun, Q., 2018. An Efficient Radio Map Updating Algorithm based on K-Means and Gaussian Process Regression. *The Journal of Navigation*, pp.1-14.
[171] Zou, H., Jin, M., Jiang, H., Xie, L. and Spanos, C.J., 2017. WinIPS: WiFi-Based Non-Intrusive Indoor Positioning System With Online Radio Map Construction and Adaptation. *IEEE Transactions on Wireless Communications*, *16*(12), pp.8118-8130.
[172] Huang, J., Millman, D., Quigley, M., Stavens, D., Thrun, S. and Aggarwal, A., 2011, May. Efficient, generalized indoor wifi graphslam. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on* (pp. 1038-1043). IEEE.
[173] Sakr, M. and El-Sheimy, N., 2017, September. Efficient Wi-Fi signal strength maps using sparse Gaussian process models. In *Indoor Positioning and Indoor Navigation (IPIN), 2017 International Conference on* (pp. 1-8). IEEE
[174] Jain, A.K., 2010. Data clustering: 50 years beyond K-means. *Pattern recognition letters*, *31*(8), pp.651-666.
[175] Fahad, A., Alshatri, N., Tari, Z., Alamri, A., Khalil, I., Zomaya, A.Y., Foufou, S. and Bouras, A., 2014. A survey of clustering algorithms for big data: Taxonomy and empirical analysis. *IEEE transactions on emerging topics in computing*, *2*(3), pp.267-279.
[176] Géron, A., 2017. *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems*. " O'Reilly Media, Inc.".
[177] Bacry, E., Bompaire, M., Gaïffas, S. and Poulsen, S., 2017. tick: a Python library for statistical learning, with a particular emphasis on time-dependent modeling. *arXiv preprint arXiv:1707.03003*.
[178] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. and Vanderplas, J., 2011. Scikit-learn: Machine learning in Python. *Journal of machine learning research*, *12*(Oct), pp.2825-2830.
[179] Comparing different clustering algorithms on toy datasets[Online] available at : http://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html#sphx-glr-auto-examples-cluster-plot-cluster-comparison-py [Accessed on 06/06/2018]

APPENDIX

[1] Syed, U B, Arslan, T., "3-Dimensional Approach to WiFi Indoor Positioning," *Proceedings of the 24th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2011)*, Portland, OR, September 2011, pp. 2861-2865

[2] Syed U B , Arslan T, Evaluation of Bluetooth low energy beacons for indoor positioning, *Proceedings of European Navigation Conference (ENC)* November 2011.

[3] Background to Bluetooth Low Energy (BLE) technology, with BLE beacons, development and implementation.

Chapter 7

Bluetooth Beacons

7.1 Introduction

This chapter introduces Bluetooth technology and design of Bluetooth low energy beacons. Bluetooth beacon design involves both hardware and software components. The main aim of current beacon development is to build a beacon that can transmit location information (latitude, longitude) which can automatically be read by a mobile device. These beacons can then be installed along with the Wi-Fi beacons. During calibration phase, when collecting data at calibration points, data from Bluetooth beacons can also be collected automatically. Data from Bluetooth beacons provide Wi-Fi location information and thereby do not have to rely on getting the Wi-Fi location information from infrastructure installers. Also, these beacons can be installed in areas where coverage of Wi-Fi beacons is not sufficient to estimate position of mobile device. Initial sections of this chapter provide a background into Bluetooth protocol and its specifications.

Bluetooth specification is maintained by Bluetooth Special Interest Group (SIG). Bluetooth SIG apart from developing the specification also manages the qualification program. Bluetooth SIG was formally announced in 1998 [91]. Some of specification versions released are v1.0, v1.1, v1.2, v2.0, v3.0, v4.0, v4.1 and latest specification v4.2 adopted in December 2014 [92] [94]. The versions till 3.0 are

considered classic Bluetooth and from v4.0 onwards, it is a combination of classic Bluetooth and low energy components. Bluetooth technology over the years has been widely integrated into phones, computers etc.

Bluetooth is a wireless technology standard for exchanging data over short distances. A Bluetooth core system consists of a radio transceiver, baseband and protocol stack. The Bluetooth radio operates between 2400MHz and 2485MHz. This frequency band is within globally unlicensed band known as 'Industrial Scientific and Medical' (ISM) 2.4GHz band. Wi-Fi technology also operates in the ISM band. The channels used by Bluetooth systems depend on the mode device is running, classic or low energy mode. Classic Bluetooth uses 79 channels with 1MHz spacing, whereas Bluetooth low energy (BLE) uses 40 channels with 2MHz spacing [93].

7.1.1 Classic Bluetooth

A Bluetooth protocol has multiple layers in the communication stack. Figure 7.1 shows various layers of a classic Bluetooth protocol. Some components of the protocol are mandatory but depending on type of device some of the components are optional. For a device to communicate with other Bluetooth devices the mandatory components must be implemented by the manufacturer. In figure 7.1[93], the components above 'RFComm protocols' are also known as profiles. Some of the profiles shown are serial port, headset and hands-free. Depending on the type of device, one or more of these profiles is supported.

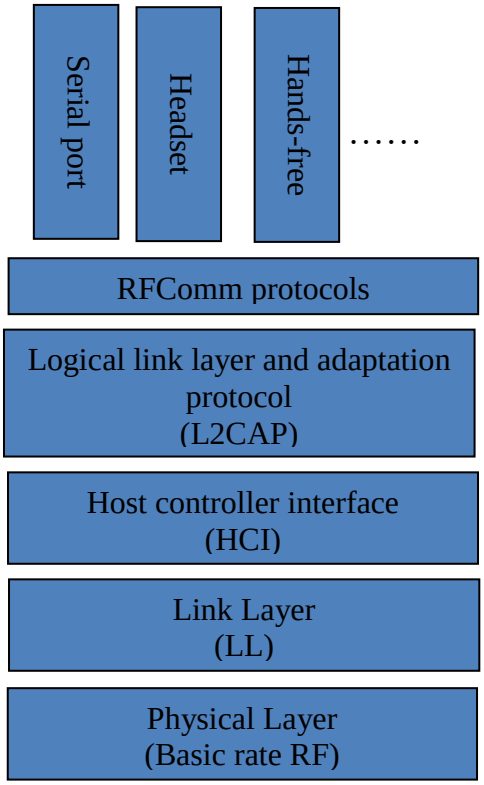


Figure 7.1 Classic Bluetooth protocol architecture

Bluetooth devices have to implement the mandatory components of the protocol to exchange data. The Bluetooth protocol requires pairing of devices before data can be exchanged. There are multiple states through which the devices go through before exchanging data. These states for a classic Bluetooth are shown in figure 7.2 [95]. A classic Bluetooth device, when powered up, gets into a standby state. In a standby state, the device can move into ‘inquiry’ or ‘page’ state where the connection is authenticated and the device moves to a connected state. Once the devices are connected, data can be transmitted in ‘transmit data’ state. After data transmission, the device can either return to a standby state or any of ‘Park’, ‘Hold’ and ‘Sniff’ modes. Usually in a classic Bluetooth mode, the data to be transmitted is continuous and large such as in case of a headset device paired with a mobile phone. Even when there is no data transmission, the devices maintain a connection which drains the battery on Bluetooth device.

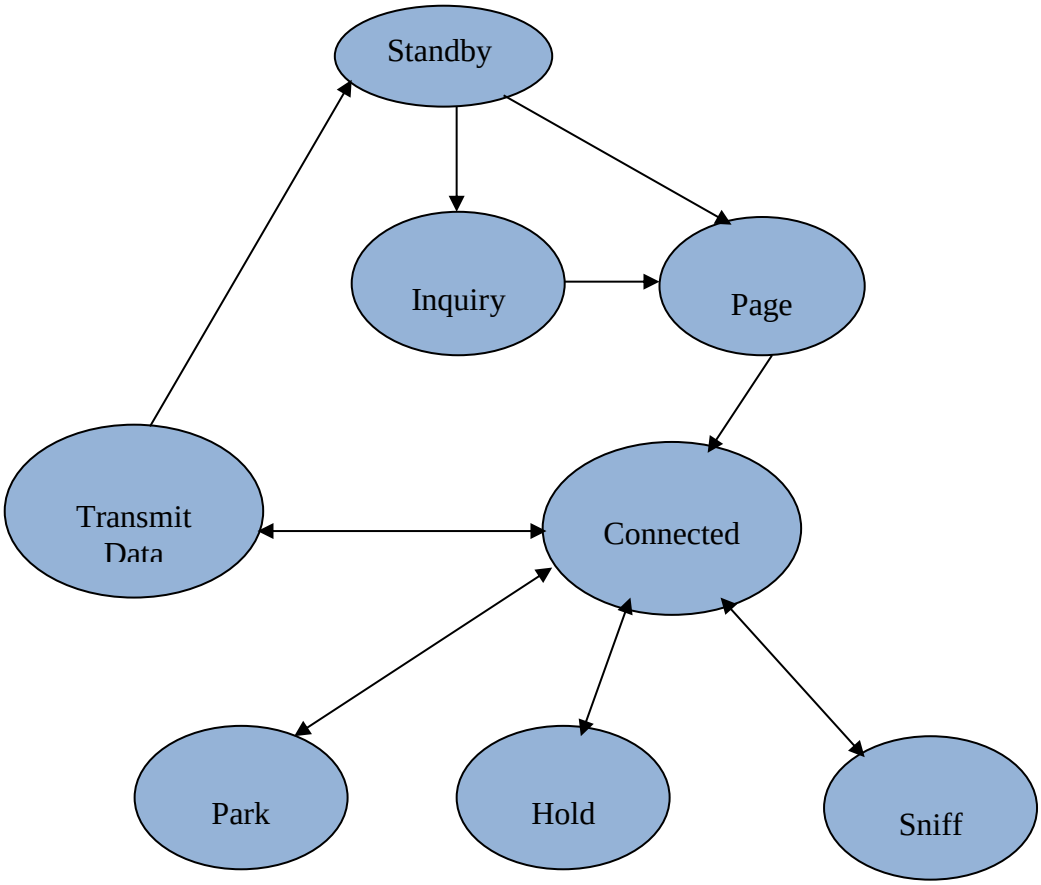


Figure 7.2 Classic Bluetooth device connection states

Bluetooth devices are divided in to three categories depending on the transmit power. Table 7.1([96] [94]) shows classes, transmit power and theoretical ranges. The actual rage of a device depends on propagation environment, line of sight and various other factors.

Class	Range (m)	Transmit power (dBm)
Class 1	100	20
Class 2	10	4
Class 3	1	0

Table 7.1 Bluetooth device classes

7.1.2 Classic Bluetooth based positioning systems

There has been extensive research into using classic Bluetooth devices for positioning purposes before adaptation of Bluetooth 4.0 in year 2010. Some works, studied signal parameters of classic Bluetooth for positioning [97] [98]. Some of the parameters that were studied are link quality (LQ), received signal strength indicator (RSSI) and transmit power level. The investigations conclude that RSSI measurement in classic Bluetooth protocol is not suitable for positioning purposes. Also, link quality depends on class of the device which require mobile device to be aware of information about the type of beacons. Hence, link quality is also not suitable for positioning purposes. However, some positioning systems were implemented using proximity techniques [99] [100], fingerprinting techniques [101] based on number of beacons detected and triangulation [102]. Hybrid positioning systems using Wi-Fi and Bluetooth technologies have also been implemented as described in [103].

7.1.3 Bluetooth low energy

Bluetooth specification was updated in 2010 to version 4.0. This updated specification adds low energy specifications to existing classic Bluetooth specifications. Bluetooth 4.0 specification introduced new range of devices which transmit short bursts of data rather than continuous data as in classic Bluetooth devices. Bluetooth 4.0 specification devices can be classified into two modes, dual mode and single mode. A dual mode device supports both classic Bluetooth and low energy specifications. Some examples of such devices are phones, laptops, tablets etc. A single mode device only supports low energy specifications. Some examples of single mode devices are remote controls, heart rate monitors etc. Single mode devices usually run on a coin cell battery and transmit only short bursts of data. Figure 7.3 and 7.4 show protocol layers of Bluetooth low energy specification for dual mode and single mode devices.

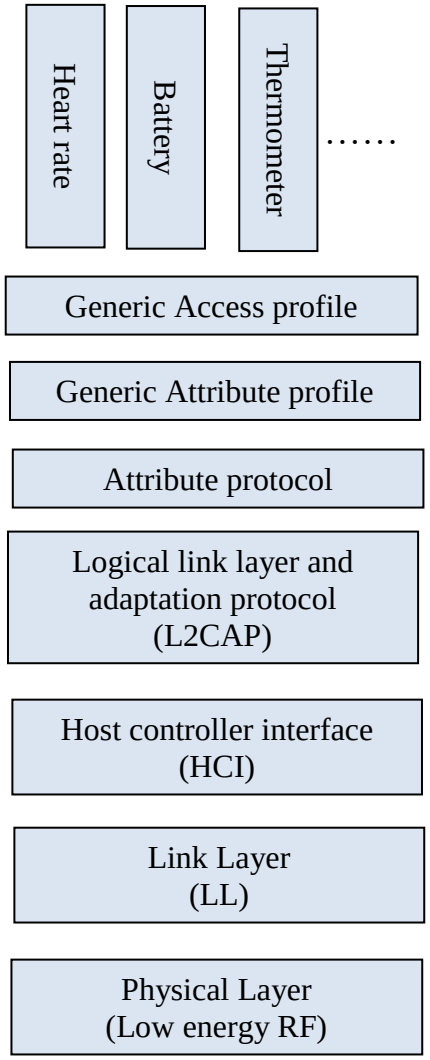


Figure 7.3 Bluetooth 4.0 protocol layers (single mode)

Dual mode devices have the capability to connect to classic Bluetooth devices as well as low energy devices. It can be noticed from figures 7.1, 7.3 and 7.4 that protocol layers for low energy specification are similar to that of classic Bluetooth. The physical layer, which represents radio hardware, is different for classic Bluetooth and low energy devices due to the change in number of channels used. Hence a dual mode device has to support both specifications to be able to communicate with classic and low energy Bluetooth devices.

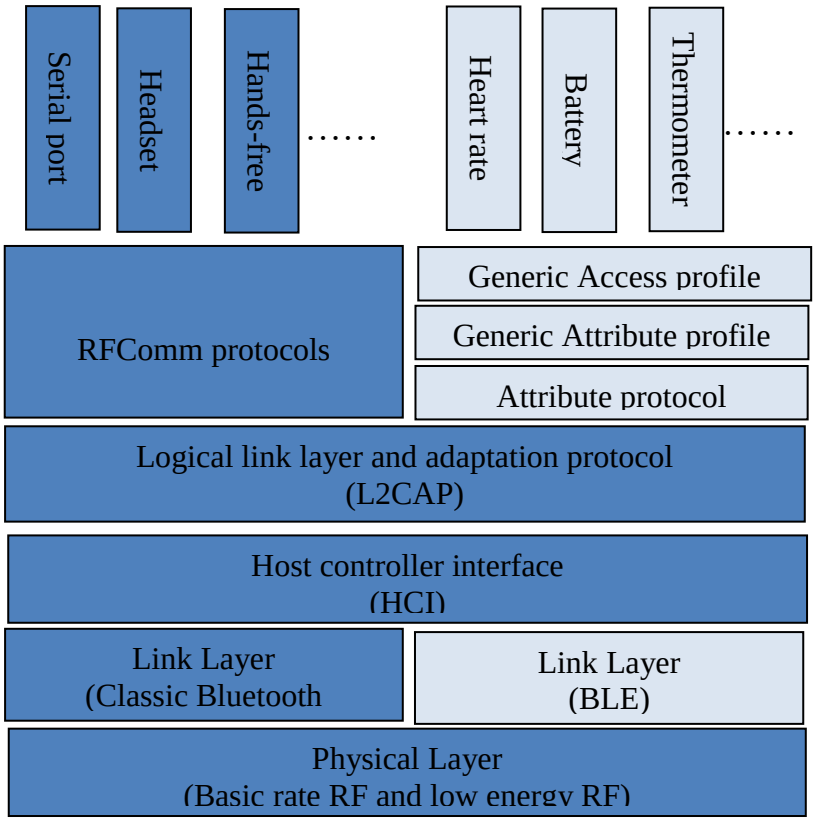


Figure 7.4 Bluetooth 4.0 protocol layers (dual mode)

Bluetooth low energy protocol uses 40 channels compared to 79 channels used by classic Bluetooth protocol. The channels used by low energy and Wi-Fi protocol are illustrated in figure 7.5. There are 40 channels used by low energy protocol of which three (37, 38 and 39) are advertisement channels and rest are data channels. Bluetooth protocol uses frequency hopping spread spectrum (FHSS) technique to reduce interference between Wi-Fi and Bluetooth devices.

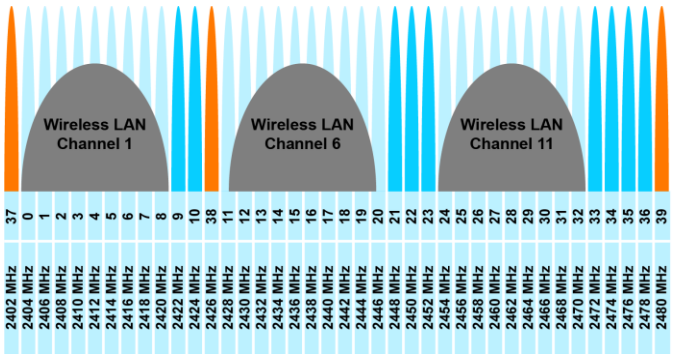


Figure 7.5 Bluetooth low energy and Wi-Fi channels in ISM band

A Bluetooth low energy device goes through multiple states for data transfer. These states are shown in figure 6.6. When a Bluetooth low energy device is powered up, it usually starts in a standby state. Depending of type of device and functionality, the device can start advertising or start scanning for other devices or can also straight away initiate a connection with a device. A device can move from either advertising or initiating state to connected state to transfer/receive data. After data transfer the devices disconnect and move back to standby state.

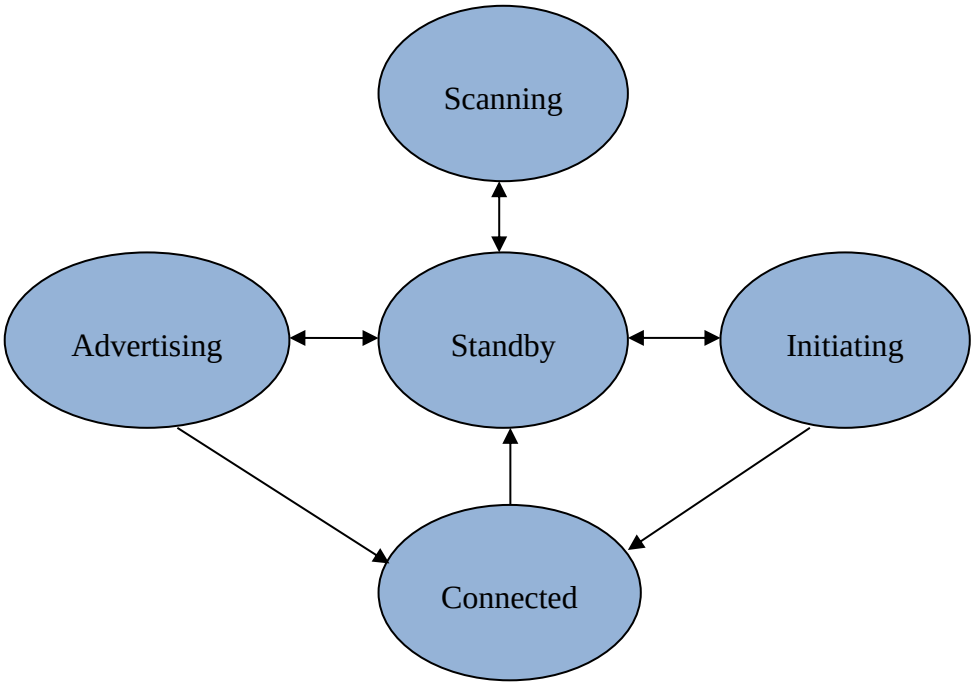


Figure 7.6 Bluetooth low energy connection states

A BLE device can also transmit data in advertising mode which can be read by all Bluetooth 4.0 compatible devices. However, the amount of data that can be transmitted as part of advertisement packets is limited to 37 bytes. A system using advertisement packets for transmitting data will be vulnerable in terms of security.

7.2 Bluetooth low energy beacon development

Profiles are implemented on top of Generic access profile in protocol stack as shown in figure 7.3. Depending on the functionality of a device, multiple profiles are implemented. For example, a heart rate monitor device can implement a heart rate profile along with battery profile which provides information about on board battery. The objective of the current BLE beacon development is to create a location profile on the beacon which can transmit MAC id of Wi-Fi beacon and location coordinates

along with additional information such as building name, floor number and room number. The proposed BLE beacon is a single mode device to which a mobile phone can connect and retrieve Wi-Fi beacon location information. The BLE beacon development involved both hardware and software development.

7.2.1 BLE beacon hardware

Bluetooth low energy protocol was part of the Bluetooth 4.0 specification released in 2010 [93] [94]. Some of the chip manufacturers to release a chip supporting single mode BLE were Texas instruments [104] and Nordic semiconductor [105]. The number of Bluetooth beacon providers have gone up after announcement of BLE protocol support on Android [106] [109] and iPhone [107] [108] platforms. Some of BLE beacon suppliers are Estimote [110], Kontakt [111] and Radius networks [112]. Majority of these beacons are based on reference designs of development kits from Texas instruments or Nordic semiconductor. The current development of BLE beacon is based on Texas instrument's CC2540 mini development kit. Developing a profile using other beacons such as Estimote is slightly complicated due to non-availability of development tools. Moreover, at the time when the current project was started, only Texas instruments development kit had tools available for a profile development.

The current beacon and profile development is based on Texas instrument's CC2540 mini development kit. This development kit has since been updated to CC2541 development kit in 2012 [113]. The main difference in the two chips is that, CC2540 supports USB (Universal Serial Bus protocol) interface whereas CC2541 does not. Also, CC2541 supports I2C (Inter Integrated Circuit protocol) whereas CC2540 does not. If the USB and I2C features on the chips are not used in an application, both CC2540 and CC2541 modules are compatible and work exactly the same. For the current location information profile development, CC2540 chip was used. However, the same profile can run on CC2541 chip as well.

The CC2540 mini development kit from Texas instruments contains three major components, a CC Debugger, key fob and USB dongle as shown in figure 7.7. The key fob is a single mode Bluetooth low energy device which can be programmed using the CC Debugger. The USB dongle can be connected to a Windows computer

acting as a serial port. This dongle enables the computer to receive and transmit data from and to the key fob.

Figure 7.7 Texas instruments CC2540 mini development kit

For the current profile implementation, the development is done on key fob and the USB dongle is not used. The Wi-Fi location information profile is flashed on to key fob using CC Debugger and tested using a mobile phone instead of a computer. Texas instruments provide a reference design for key fob which has been modified for current design of beacon.

7.2.2 BLE Beacon software

The current objective is to design a Wi-Fi beacon location information profile which can transmit coordinates of a Wi-Fi beacon and additional information such as building name, floor number, and the room in which beacon is located. A profile operates on top of Generic Access Profile (GAP) layer in the communication layers as shown in figure 7.3. Profile implementation requires understanding of the top two layers of the protocol, Generic Access Profile (GAP) and Generic Attribute Profile (GATT).

Generic Access Profile (GAP) is responsible for the device's access roles and procedures such as device discovery, link establishment, link termination and device configuration. A Bluetooth 4.0 device operates in one of the following roles; Broadcaster, Observer, Peripheral and Central. A device can support more than one role, depending on the application. In Broadcaster role, device can only advertise and is non-connectable, where as in Observer role, the device can only scan for

advertisements but cannot initiate connections. A device in Peripheral role is connectable and acts as a slave in a data connection. The final role in which a device can operate is Central mode, in this role a device scans for advertisements and can initiate a connection. Bluetooth 4.0 allows for a device to support multiple roles. Central role is supported by devices such as phones and laptops which can connect to single mode devices such as heart rate monitors.

Generic Attribute Profile (GATT) layer of the protocol stack is designed to be used by the application for data transfer between connected devices. Devices connected for data transfer can be either GATT client or GATT server. A GATT server is the device that contains data to be read and GATT client is the device that reads data from GATT server. In the current implementation, Key fob is GATT server and mobile phone is GATT client. This client-server architecture is shown in figure 7.8 [114].

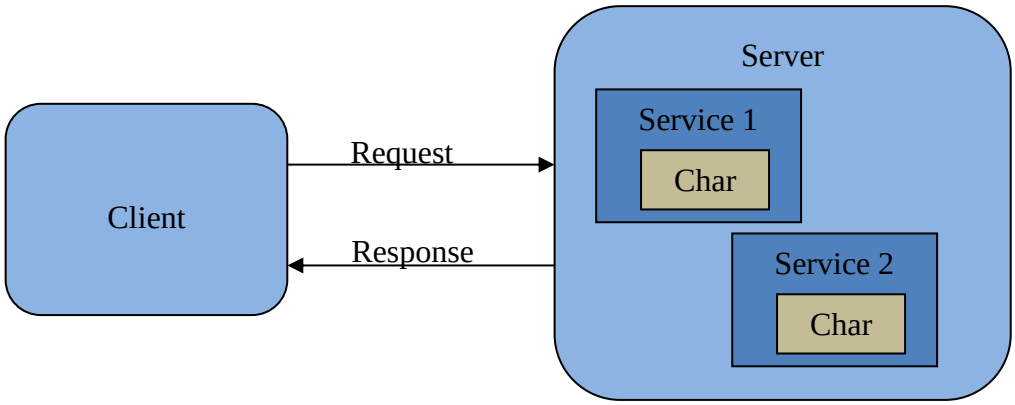


Figure 7.8 GATT Client-Server Architecture

The data on server is grouped into services and each service has data exposed to the client as ‘characteristics’. Each service can have multiple related characteristics, for example a Wi-Fi beacon location information service can have characteristics such as ‘MAC id’, ‘Latitude’, ‘Longitude’, ‘Building Name’, ‘Floor number ‘and ‘Room number/Name’. The characteristic values along with their properties are stored on the GATT server in an ‘Attribute Table’. Each attribute in the ‘Attribute table’ has three properties along with the value itself. These properties are ‘Handle’, ‘Type’ and ‘Permissions’. Handle is the attribute’s address in the table and each attribute has a unique handle. The ‘Type’ indicates what the data represents, for example the Wi-Fi beacon location information service has characteristics

‘Latitude’ and ‘Longitude’ which have values but ‘Type’ identifies what these values represent as in latitude or longitude etc. ‘Permissions’ determines how the GATT client can access data on the server. For example, some fields on the server can only be read while some can be written to as well. These access modes are determined by the ‘Permissions’ property of each attribute. The structure of Wi-Fi beacon location information profile is shown in table 7.2 which shows the characteristics, their UUIDs (Universal Unique Identifier) in hexadecimal format and their values. All the characteristics in this profile have ‘read only’ access. All characteristics have a 128-bit UUID which is usually assigned by Bluetooth SIG approved. However, in current development, the UUIDs are not Bluetooth SIG approved and are specific for the profile being developed. The latitude and longitude values have precision up to seven decimals and when transmitting the values, decimal is dropped so that the application does not have to deal with floating point values. When these values are received on the mobile end, they are divided with ‘100000000’ so that the original latitude and longitude values can be retrieved.

Characteristic name	UUID	value
Wi-Fi location information service	0000fee0-0000-1000-8000-00805f9b34fb	Wi-Fi location information service
Latitude	0000fee1-0000-1000-8000-00805f9b34fb	559228660
Longitude	0000fee2-0000-1000-8000-00805f9b34fb	-31718820
Building name	0000fee3-0000-1000-8000-00805f9b34fb	SANDERSON
Floor number	0000fee4-0000-1000-8000-00805f9b34fb	0
Room number	0000fee5-0000-1000-8000-00805f9b34fb	1
Room name	0000fee6-0000-1000-8000-00805f9b34fb	Classroom1
Wi-Fi beacon MAC Id	0000fee7-0000-1000-8000-00805f9b34fb	54:78:1A:21:3B:C0

Table 7.2 Wi-Fi Location information service attributes

Texas instruments CC2540 development kit provides a windows application called ‘BTool’ [115] which can be used for testing the applications and profiles developed. The tool displays data transferred between laptop and key fob. Figure 7.9 shows the screenshot of data transaction when a scan is initiated. The section highlighted in green is data sent from the laptop and the following sections in blue are data received from key fob. Access to the raw data transmitted and received is helpful for troubleshooting any issues when developing a new profile.

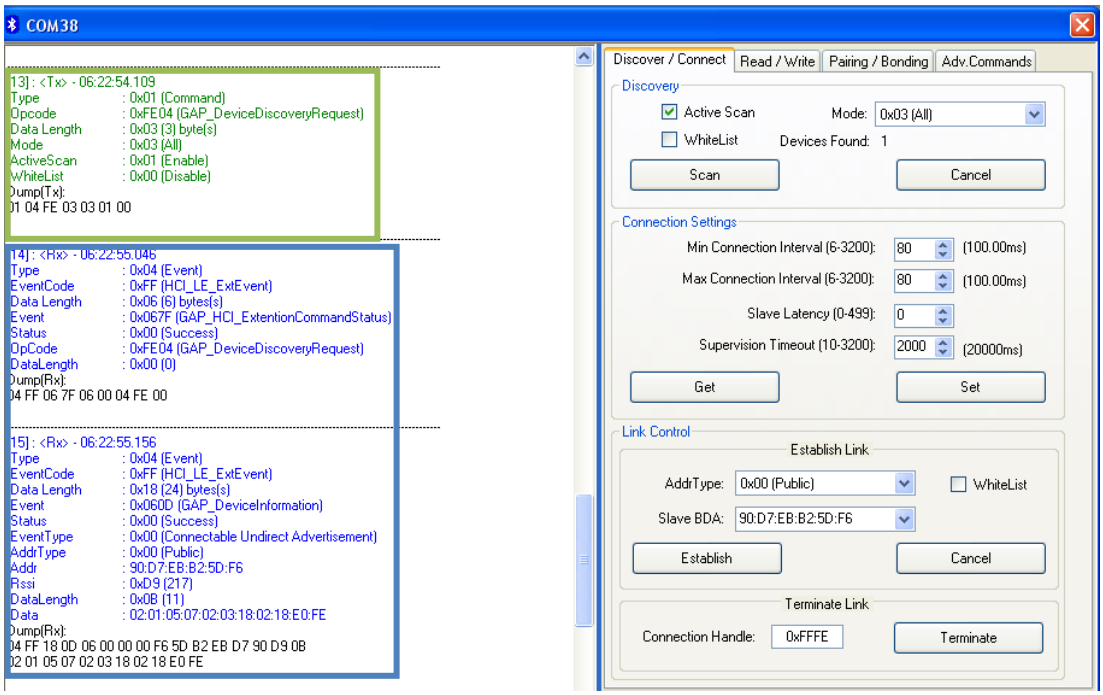


Figure 7.9 Screenshot of BTool application

Developing a new BLE profile requires a development environment and CC2540 requires ‘Embedded workbench for 8051’ from IAR software. An evaluation version can be downloaded from the URL ‘https://www.iar.com/iar-embedded-workbench/downloads/’. A screenshot of the application is shown in figure 7.10.

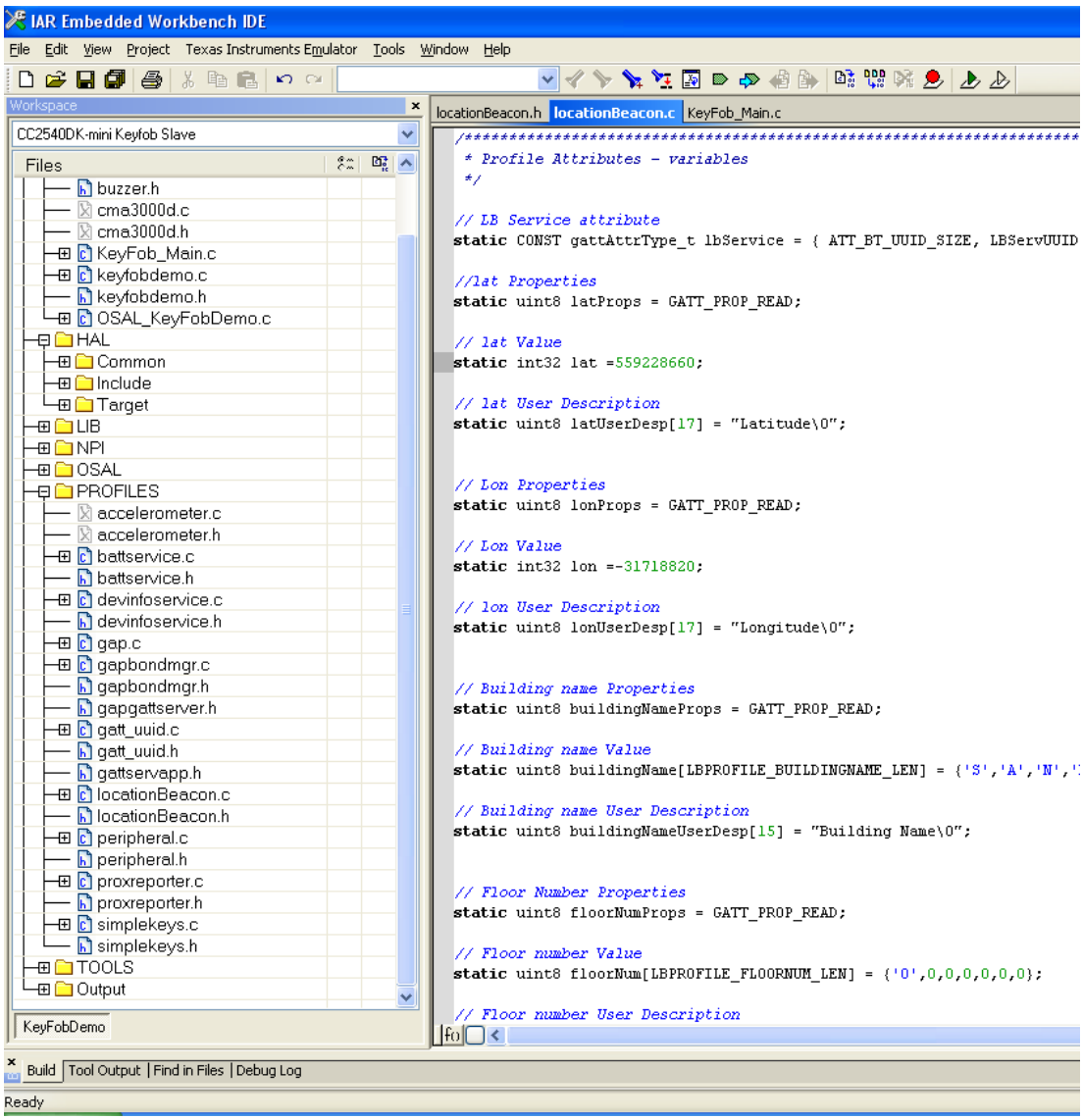


Figure 7.10 IAR Embedded Workbench

The source code of demo application named ‘KeyFobDemo’ is provided along with the development kit. The ‘KeyFobDemo’ application implements multiple profiles such as battery service, device information service, proximity service etc. A new profile can be flashed and debugged onto key fob using the CC Debugger through IAR embedded workbench.

Apart from the development on beacon, an application is to be developed on the phone which reads data from BLE beacon. Android studio was used for application development. Sample application to scan and connect to Bluetooth low energy is provided as part of Android studio. The sample application used as basis for the current development is ‘BluetoothLeGatt’.

7.3 BLE signal characteristics

Apart from hardware and software for the beacons, another aspect to study is characteristics of BLE radio signals. The two main BLE radio signal characteristics studied in this section are RSS and transmit power. These characteristics determine the range of beacons and stability of signal over distance. A classic Bluetooth device cannot change its transmit power. This section explores the possibility of updating transmit power of BLE device. Also, in this section, RSS variation is studied in two scenarios, first the variation of RSS at a fixed distance over time and secondly, variation of RSS with distance.

Texas instruments provide various APIs (Application Programming Interface) to program key fob hardware. APIs provided are for different levels of the protocol stack illustrated in figures 7.3 and 7.4. The HCI (Host controller interface) layer of the protocol stack acts as an interface between host and controller. One of the APIs to control transmit power is 'HCI_EXT_SetTxPowerCmd'. This API takes transmit power to be set as a parameter and returns status of execution. The possible input parameters are displayed in the table 7.3. The 'HCI_EXT_SetTxPowerCmd' returns '0X00' on successful execution.

Input parameter value	Parameter description
0X00	HCI_EXT_TX_POWER_MINUS_23_DBM
0X01	HCI_EXT_TX_POWER_MINUS_6_DBM
0X02	HCI_EXT_TX_POWER_0_DBM
0X03	HCI_EXT_TX_POWER_4_DBM

Table 7.3 HCI command parameters

'HCI_EXT_SetTxPowerCmd' allows control over the transmit power at which the device operates and thereby control over the range the device can transmit.

The other characteristic to be investigated is the variation of RSS. Initial experiment conducted is to study signal variation at 1m distance between BLE beacon and laptop. RSS was measured every second for over a minute and figure 7.11 shows plot of the data collected. It can be noticed that measured RSS is not constant and fluctuates up to 15dB, even though the distance is constant and laptop is stationary. This behaviour is similar to Wi-Fi beacons as shown in figure 2.9 which shows variation of RSS measured at a distance of 5m from Wi-Fi beacon.

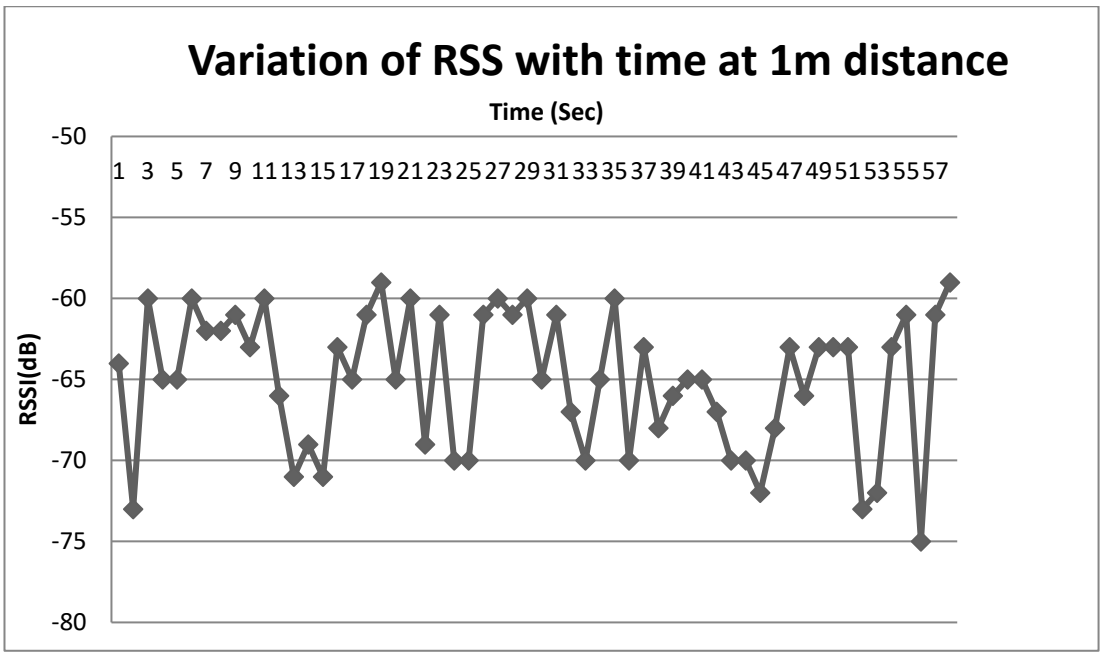


Figure 7.11 Plot of RSS at 1m distance from beacon over one minute

The next experiment was to evaluate variation of RSS with distance. For this experiment, RSS values were collected at multiple locations and data was plotted. A BLE beacon was placed in a room of 13m × 6m and RSS was collected at every one metre. Figure 7.12 shows plot of measured RSS and distance between beacon and laptop. The plot shows that RSS decreases exponentially with distance.

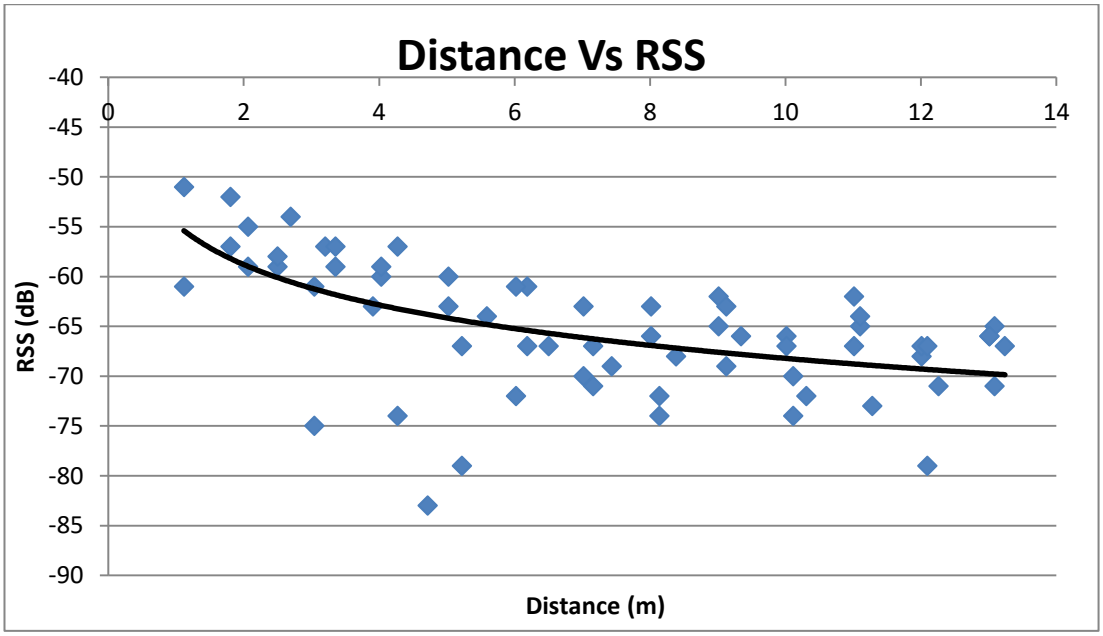


Figure 7.12 Distance Vs RSS plot of BLE beacon

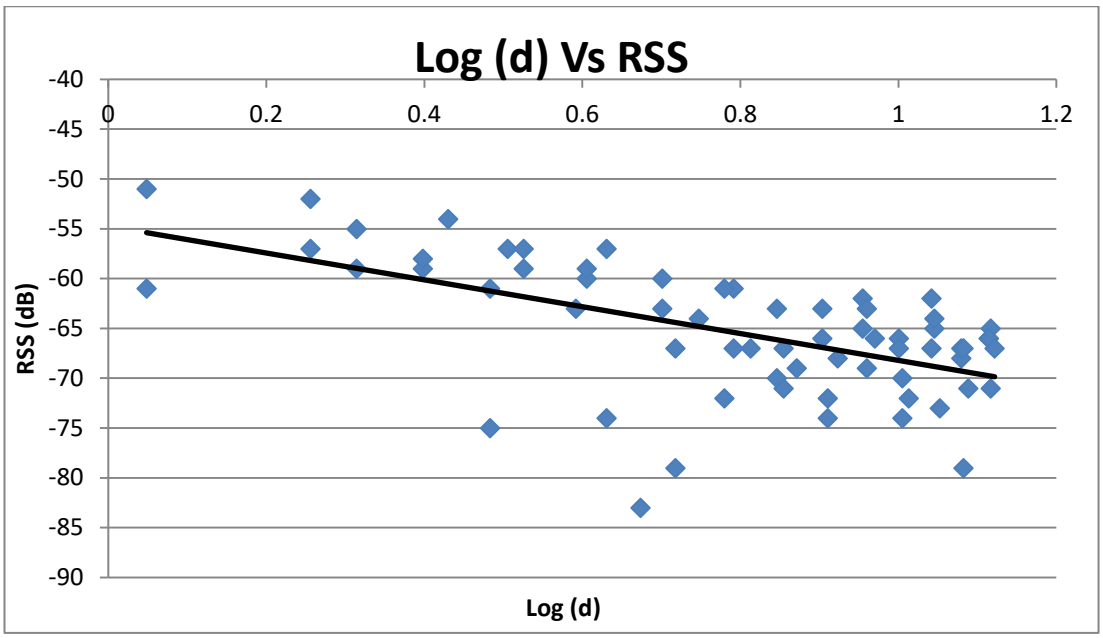


Figure 7.13 Log(d) Vs RSS plot of BLE beacon

Another plot of log distance and RSS is shown in figure 7.13 which shows that RSS varies linearly with log-distance. The plots show that BLE signal behaves similar to other radio signal propagation such as Wi-Fi.

7.4 BLE beacon implementation

This section details hardware and software implementation of BLE beacon. Initially a PCB design using Protel is detailed, followed by BLE profile development using IAR embedded work bench. An Android application development using Android studio is also detailed.

7.4.1 BLE beacon hardware implementation

Texas instruments provided a reference design for key fob PCB (Printed Circuit Board) which was used as reference to develop a custom BLE beacon circuit board. Figure 7.14 shows modified layout of new circuit board. The new board follows reference design and makes modifications to battery size and components size, was designed using Protel PCB design software.

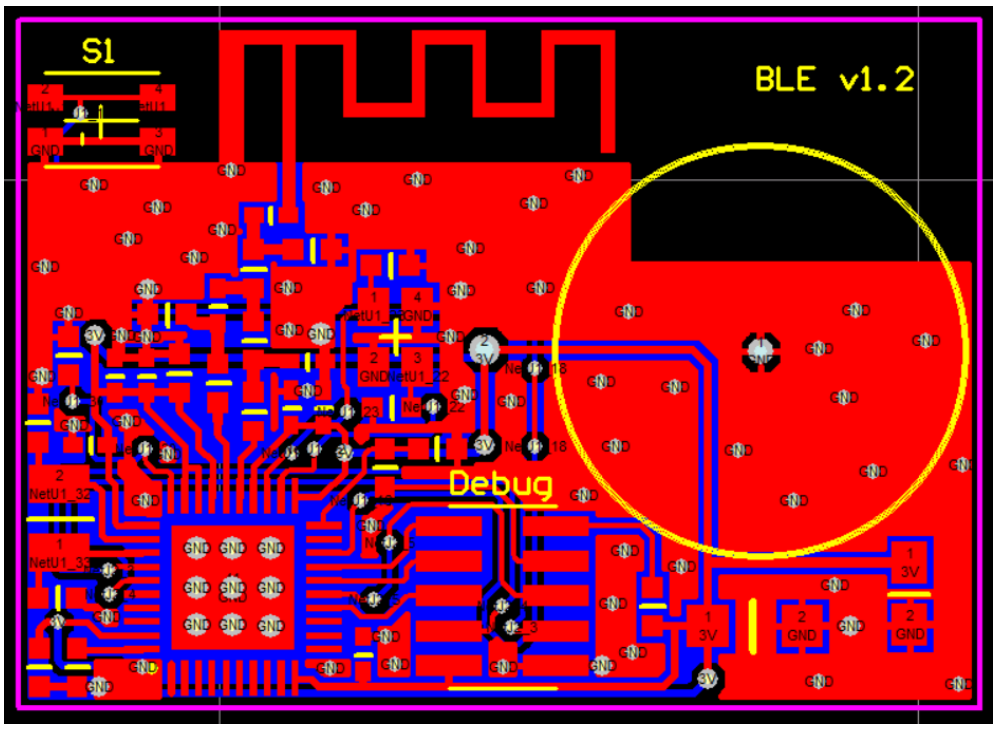


Figure 7.14 BLE beacon layout

On the new board design some of the components which are not essential for a beacon were removed to reduce the size and cost of the board. Some of the components on the reference design, but removed from the new board are LED, a switch, and test pins. Since some of the components are removed, profiles which need switches or LED will not be supported. But for current implementation of a Location information profile, these components are not used. However, the debug pins were retained which allows reprogramming of the board with updated BLE protocol stack. Figure 7.15 shows an image of the fabricated BLE beacon.

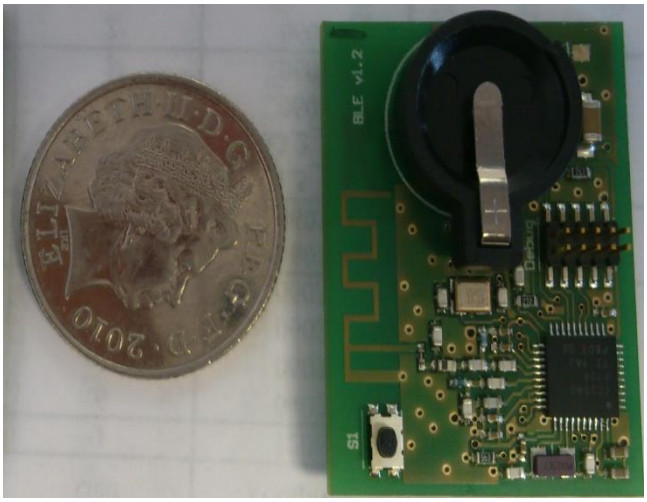


Figure 7.15 BLE beacon

7.4.2 Beacon software implementation

The software implementation for the beacon involves development on fabricated (CC2540) beacon and also on mobile phone. IAR embedded workbench environment is used for development on the beacon and Android studio is used for development on the phone.

7.4.2.1 BLE profile development

Wi-Fi beacon location information profile code was added to the sample application ‘KeyFobDemo’. Opening the source code for ‘KeyFobDemo’ using IAR embedded workbench shows the list of application code files and profile code files. A screenshot of the list of files is shown in figure 7.16, which shows Wi-Fi beacon location information profile source files name named ‘locationBeacon.h’ and ‘locationBeacon.c’ highlighted in green.

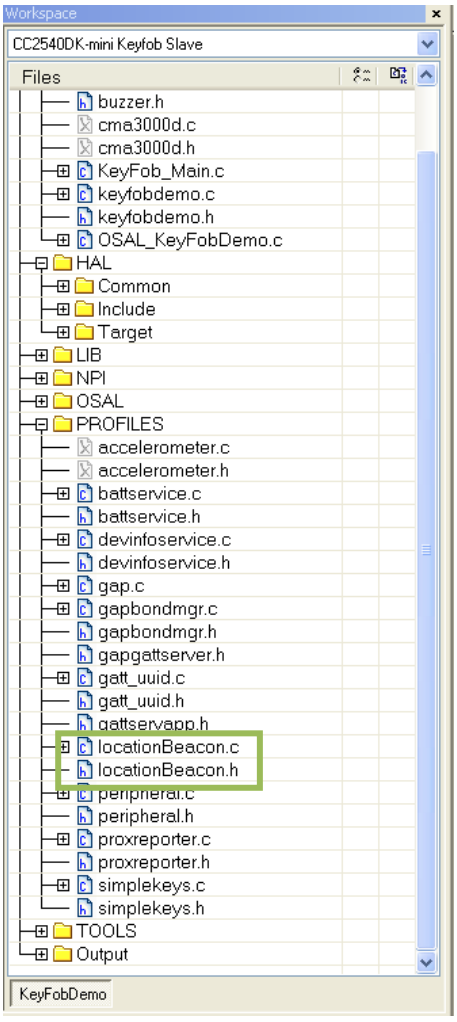


Figure 7.16 KeyFobDemo application source files with Wi-Fi beacon location information profile

In ‘locationBeacon.h’ file, the UUID of characteristics and length of characteristics used in the profile are declared as shown in figure 7.17. The section highlighted in green shows first 32 bits of UUIDs and rest of the bits are stored as part of Bluetooth stack. The section highlighted in red shows length of characteristics used in the profile.

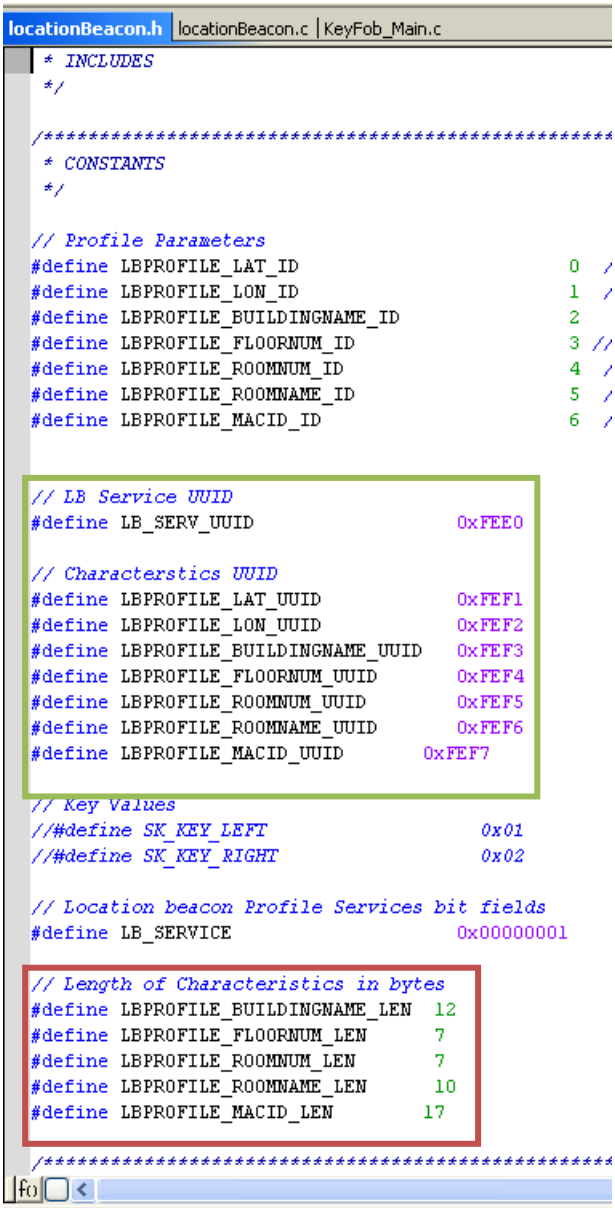


Figure 7.17 Screenshot of LocationBeacon.h file

The values for characteristics are assigned in ‘LocationBeacon.c’ file. The Wi-Fi beacon location information profile’s seven characteristics, values and types listed in table 7.2. Figures 7.18 and 7.19 show screenshots of ‘LocationBeacon.c’ file and highlights profile characteristics.

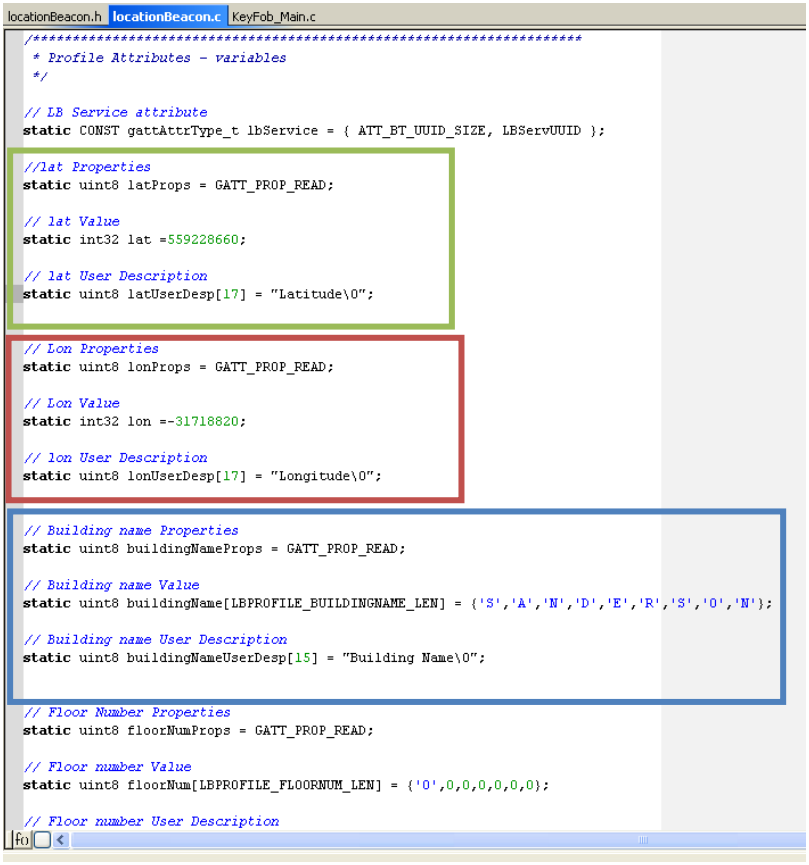


Figure 7.18 Profile characteristics assignment 1

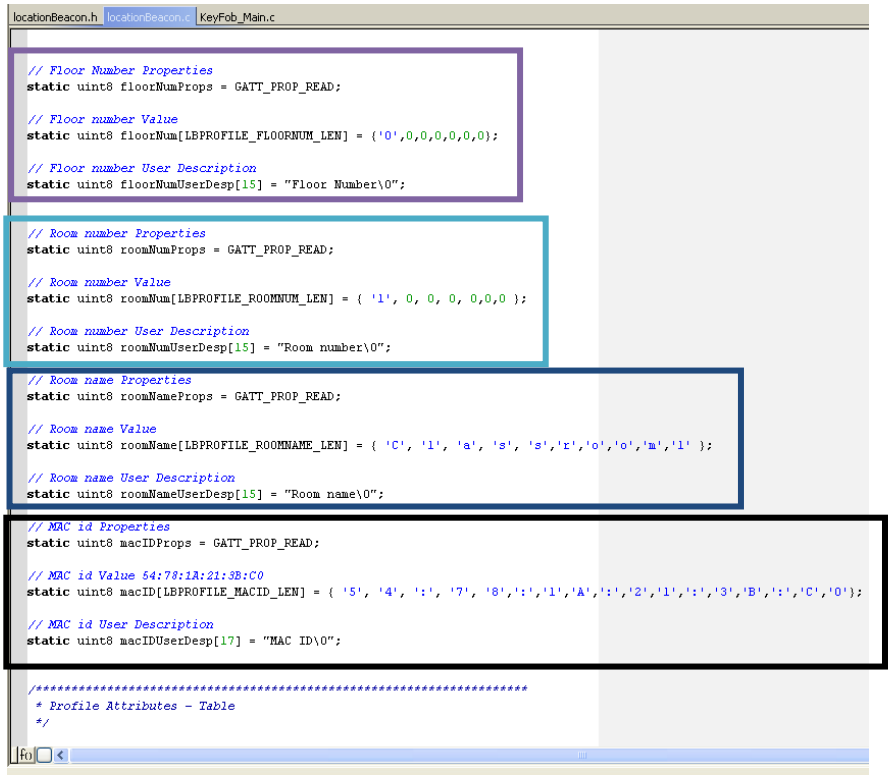


Figure 7.19 Profile characteristics assignment 2

The function ‘lb_ReadAttrCB’ highlighted in green is called when one of the characteristics on the key fob is read by mobile device. The access permissions for characteristic are first checked to ensure the characteristic being read has appropriate permissions. Only if the characteristic has read permission, a switch – case based on UUID returns characteristic value, as highlighted in red in figure 7.20.

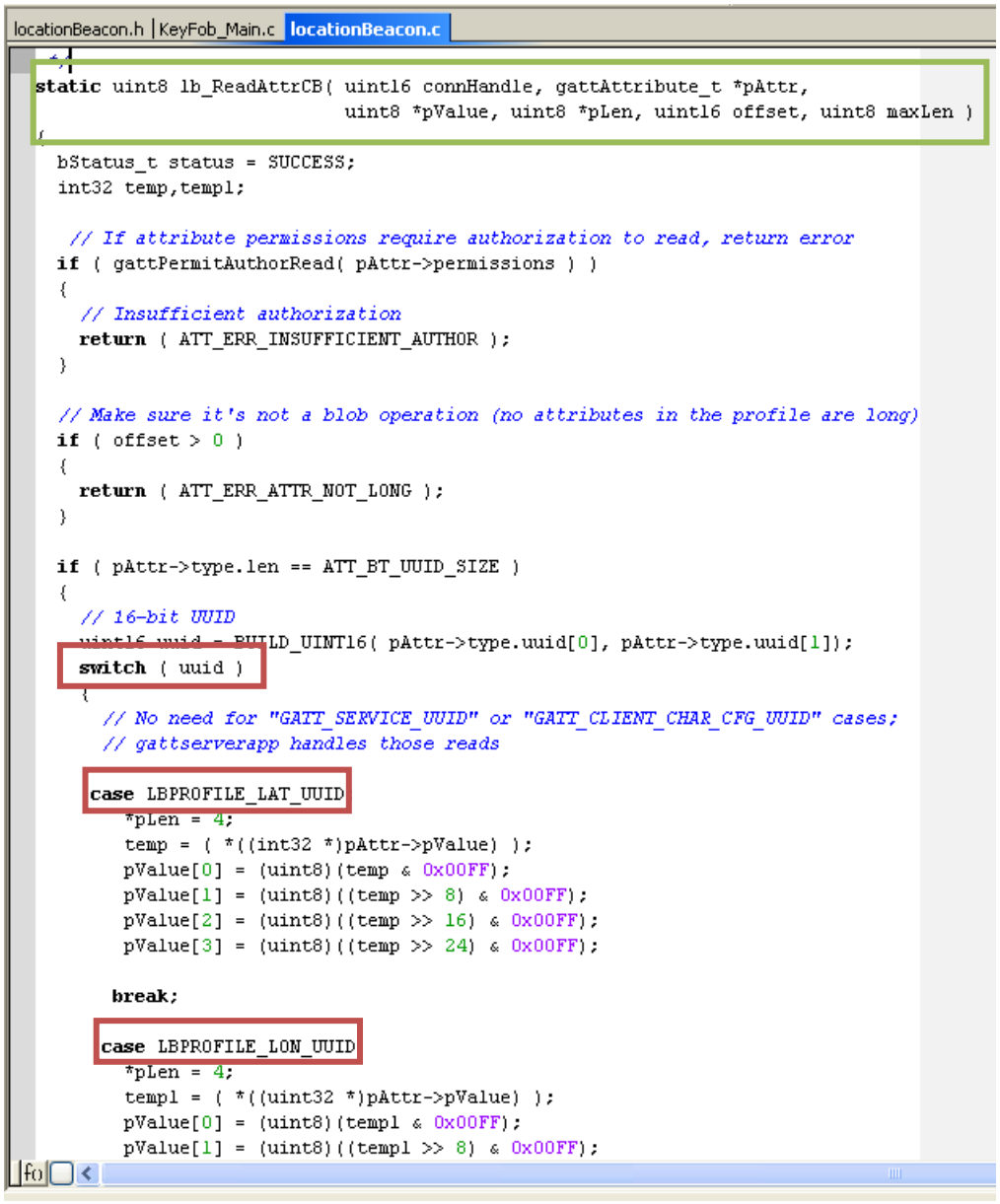


Figure 7.20 Call back function to read Characteristics from key fob

The application on the beacon is updated by flashing the key fob with the new profile. The key fob is can be flashed from IAR embedded workbench by using the button on the tool bar highlighted in figure 7.21.

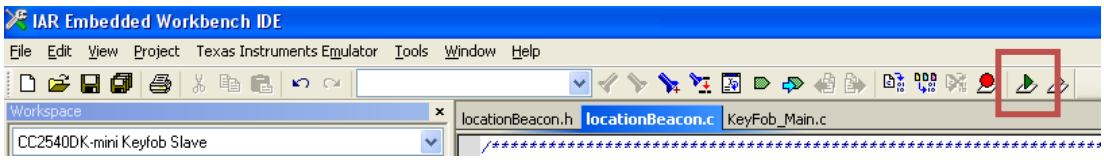


Figure 7.21 IAR embedded workbench tool bar

The beacon is connected to a computer through CC Debugger as shown in figure 7.22. The green LED on CC Debugger indicates successful detection of beacon and ready to be flashed.

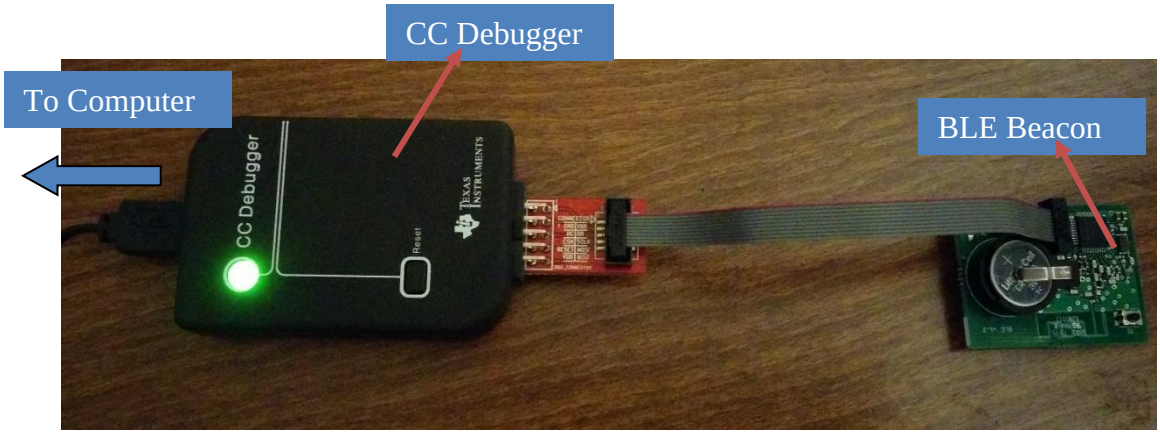


Figure 6.22 Beacon connected to computer via CC Debugger

7.4.2.2 BLE application development on Android

Android mobile device was used for application development which can access the Wi-Fi location information on BLE beacon. Android studio [116] was used for current application development. A Bluetooth low energy sample application is included in android studio and for the current development this sample application is used as a base. The sample application 'BluetoothLeGatt' scans for Bluetooth low energy devices and can read data after connecting to BLE beacon. The source code of sample application has four files, 'BluetoothLeService.java', 'DeviceControlActivity.java', DeviceScanActivity.java' and 'SampleGattAttributes.java'. Figure 7.23 shows declaration of seven characteristic UUIDs in 'SampleGattAttributes.java' file, which are used to identify characteristics when read from BLE beacon.

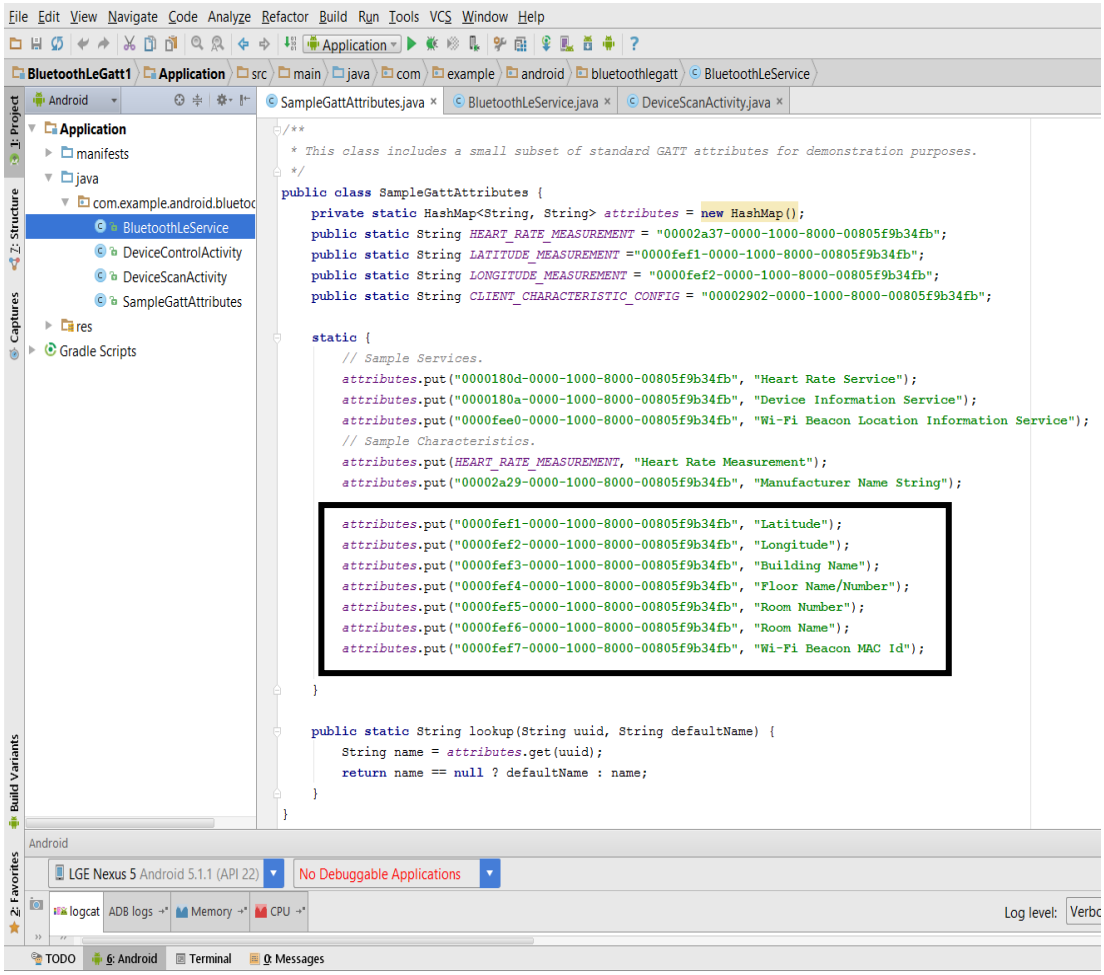


Figure 7.23 Characteristics UUID declarations for mobile application

The latitude and longitude values read from the key fob are divided by '1000000' to adjust for dropping the decimal point on beacon. The conversion code is shown in figure 7.24 which is a screenshot of file 'BluetoothLEService'.

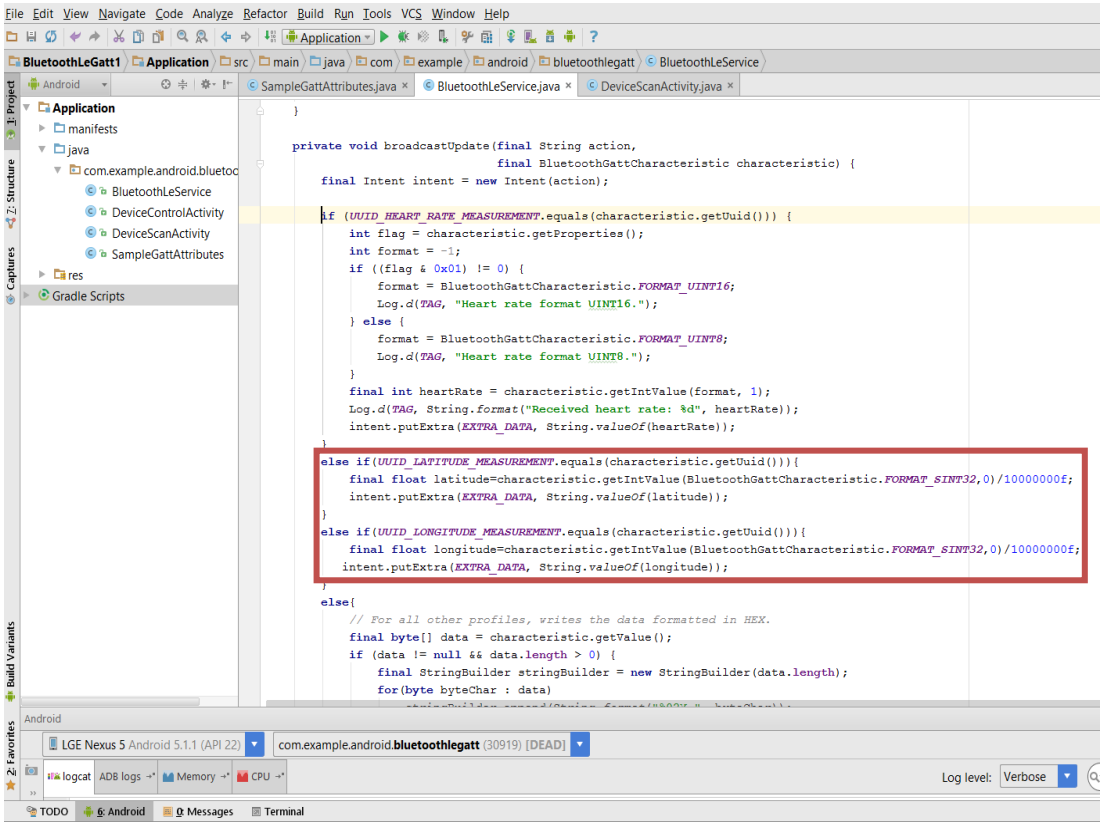


Figure 7.24 Screenshot of 'BletoothLeService'

The android application scans for nearby beacons and displays a list of detected beacons as shown in figure 7.25. Tapping on beacon name (keyfobdemo) from the list connects to the beacon. The screen shows MAC id of the beacon, state (connected) and list of services on the beacon. One of the services on the list is 'Wi-Fi Beacon Location Information Service'. Tapping on the service, reads data from beacon and displays the list of characteristics supported by the service. Tapping on each characteristic retrieves data from the beacon and is displayed on the screen as shown in figures 7.26, 7.27 and 7.28.

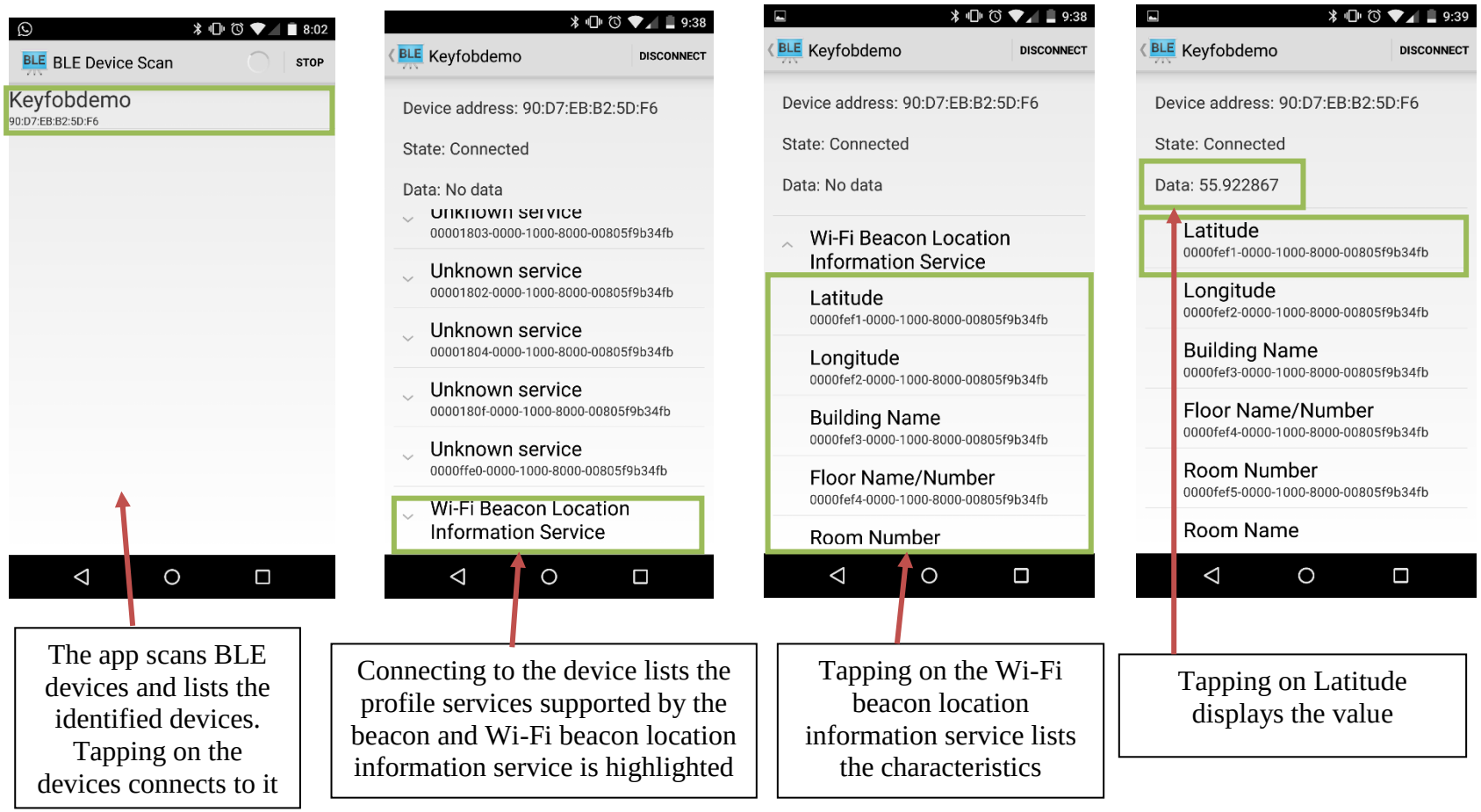


Figure 7.25 Screenshots of android application reading data from BLE beacon

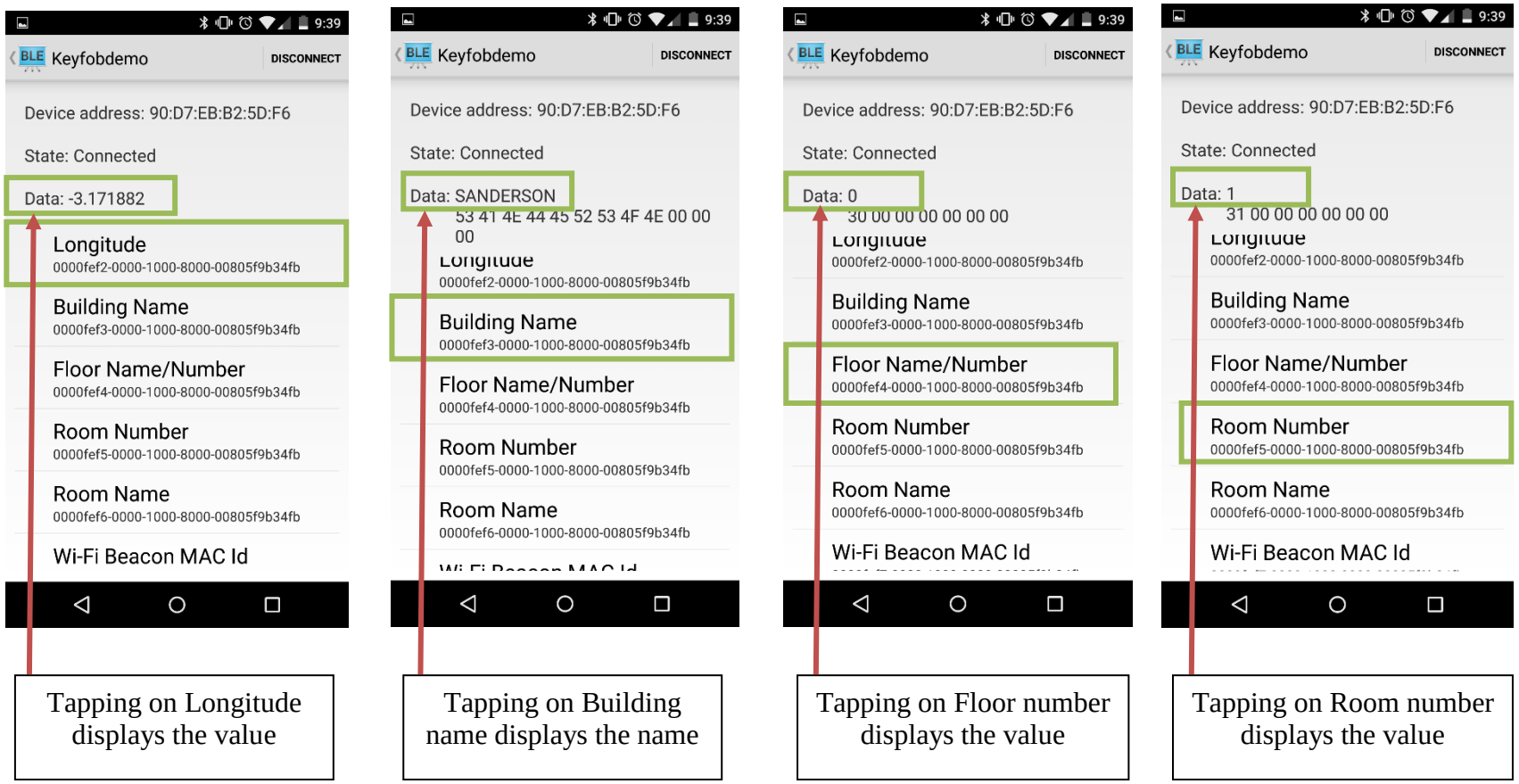


Figure 7.26 Screenshots of android application reading data from BLE beacon

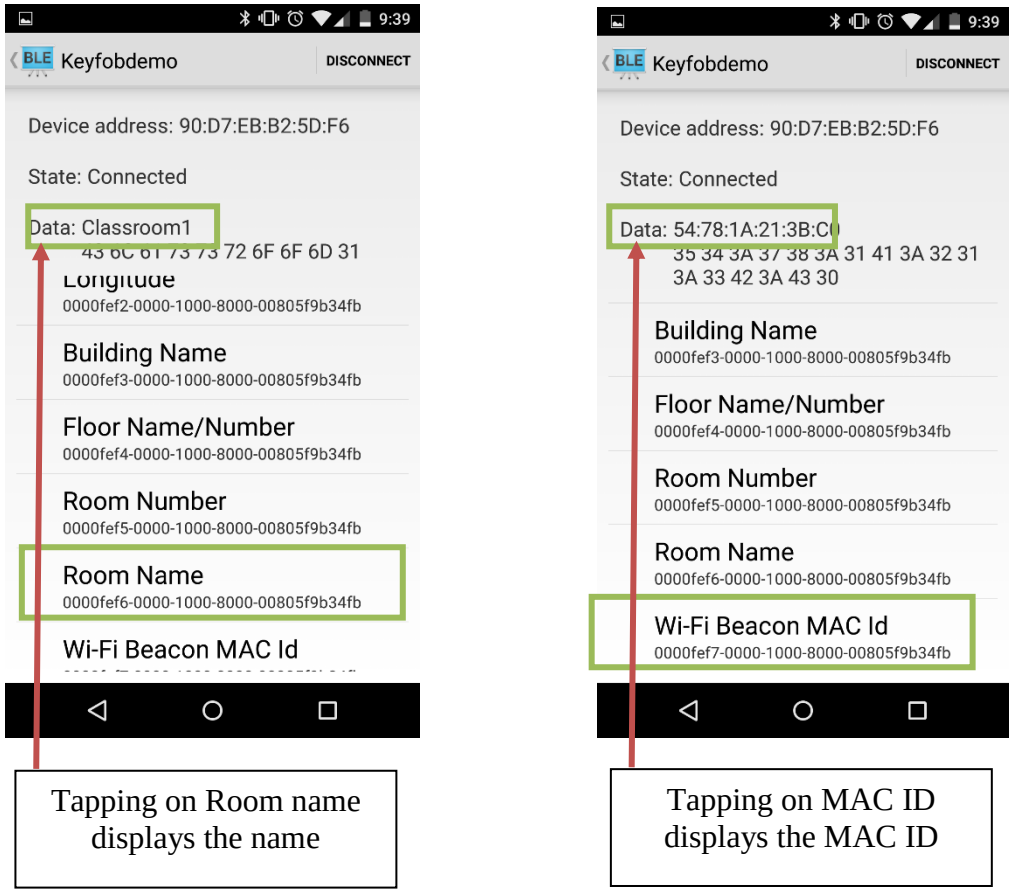


Figure 7.27 Screenshots of android application reading data from BLE beacon

6.5 Conclusion

Beacon hardware and software were designed and successfully implemented in this chapter. The fabricated hardware was flashed with Wi-Fi beacon location information service. An android application was successfully implemented to read Wi-Fi beacon location information. The current android application proves that BLE beacons can be used to collect Wi-Fi beacon location information successfully. The current Wi-Fi beacon location information profile includes other characteristics such as building name, floor number, and room number which are not used in current algorithm. However, the additional information would be helpful in providing location specific information on a scaled-up system.

In current implementation the android application collects Wi-Fi location data is an independent application. The proposed scalable system aims to integrate this Wi-Fi location data collection application into an application which is used to collect calibration data. Thereby reducing the effort and resources required to collect Wi-Fi beacon location data.

Furthermore, the current BLE beacons can also altered to transmit its own location. The BLE beacons can be installed in poor Wi-Fi coverage areas. This would be particularly helpful in cases where an area of interest only has poor existing Wi-Fi infrastructure. Instead of having to install expensive Wi-Fi beacons in such scenarios, these small BLE beacons which can run off a battery can be installed.